Incremental Adaptive Organization for a Satellite Constellation

Grégory Bonnet and Catherine Tessier

ONERA-DCSD, 2 avenue Edouard Belin BP 74025 31055 Toulouse Cedex 4, France
gregory.bonnet@onera.fr, catherine.tessier@onera.fr

Abstract. Physical agents, such as robots, are generally constrained in their communication capabilities. In a multi-agent system composed of physical agents, these constraints have a strong influence on the organization and the coordination mechanisms. Our multi-agent system is a satellite constellation for which we propose a collaboration method based on an incremental coalition formation in order to optimize individual plans so as to satisfy collective objectives. This involves a communication protocol, a common knowledge notion and two coordination mechanisms: (1) an incentive to join coalitions and (2) coalition minimization. Results on a simulated satellite constellation are presented and discussed.

1 Introduction

In the agent literature, and more precisely in a multi-agent context, most of the coordination mechanisms deal with *software agents* or *social agents* that have high communication and reasoning capabilities. Coordination based on norms [6], contracts [14] or organizations [4, 9] are considered. However in real-world applications, communication constraints have to be considered in order to share information and to coordinate.

As far as *physical agents* such as robots or satellites are concerned, the environment has a major impact on coordination due to the physical constraints that weigh on the agents. Indeed, on the one hand, an agent cannot always communicate with another agent or the communication possibilites are restricted to short time intervals. On the other hand, an agent cannot always wait until the coordination process terminates before acting. All these constraints are present in space applications.

In the space domain, autonomous satellite constellations (i.e. closed networks of satellites) allow to consider joint activities and ensure functional robustness [5]. We consider a set of 3 to 20 satellites placed in low orbit around the Earth to take pictures of the ground. Ground stations send the satellites asynchronous observation requests with various priorities. Satellites are also equipped with a detection instrument that allows areas of interest to be detected and on-board observation requests to be generated. As each satellite is equipped with a single observation instrument with use constraints, too close requests cannot be realized by the same satellite. Likewise each satellite is constrained in memory resources and can realize only a given number of requests before downloading[1]. Finally, the orbits of the satellites cross around the poles: two (or more) satellites that meet in the polar areas can communicate *via* InterSatellite Links (ISL) without any ground intervention. So the satellites can communicate from time to time in order to share information and coordinate.

Centralized planning is not considered because (1) the aim of future space applications is to avoid using ground stations as much as possible (operating a ground station is expensive); (2) the asynchronous generation of new requests by each satellite prevents having a centralized view of the problem and therefore a centralized resolution.

Consequently, the problem we focus on is a decentralized task allocation problem in a multi-agent system with new tasks arriving asynchronously and intermittent communications. Each satellite (each agent) builds and revises a task plan such that the number of tasks realized by the constellation is the highest possible, they are realized as soon as possible, the number of redundancies[2] is the lowest possible and the number of high priority tasks that are not realized is the lowest possible. Notice that these constraints are not necessarily compatible with each other. The communication problem was firstly addressed in [3]. In this paper the allocation problem is addressed with an online incremental dynamic organization mechanism in three steps:

1. agents plan individually ;
2. agents communicate in order to build a common knowledge ;
3. agents build and revise coalitions that influence their individual plans.

2 The agents

2.1 The multi-agent system structure

The constellation is a multi-agent system defined as follows:

Definition 1 (Constellation) *The* constellation $\mathcal{S}$ *is a triplet* $\langle \mathcal{A}, \mathbb{T}, \mathit{Vicinity} \rangle$ *with* $\mathcal{A} = \{a_1 \dots a_n\}$ *the set of* n *agents representing the* n *satellites,* $\mathbb{T} \subset \mathbb{N}^+$ *a set of dates defining a common clock and* $\mathit{Vicinity} : \mathcal{A} \times \mathbb{T} \mapsto 2^{\mathcal{A}}$ *a symmetric non transitive relation specifying for a given agent and a given date the set of agents with which it can communicate at that date (acquaintance model). Vicinity represents the temporal windows when the satellites meet; it is calculated from the satellite orbits, which are periodic.*

[1] Downloading consists in transferring data to a ground station (i.e. the pictures taken when a task is realized).

[2] There is a redundancy when two different agents realize the same task whereas only one would have been sufficient.

Definition 2 (Periodicity) *Let $\mathcal{S}$ be a constellation and $\{p_1 \dots p_n\}$ the set of the orbital cycle durations $p_i \in \mathbb{T}$ of agents $a_i \in \mathbb{A}$. The Vicinity period $\mathring{p} \in \mathbb{T}$ is the* lowest common multiple *of set $\{p_1 \dots p_n\}$.*

We define communication within the constellation:

Definition 3 (Communication) *Let $\mathcal{S}$ be a constellation and a_i, $a_j \in \mathcal{A}$:*

- *(Figure 1) Agent a_i can* communicate directly *with agent a_j iff $\exists\ \tau$ within $\mathring{p}$ such as $a_j \in$ Vicinity(a_i, τ);*
- *(Figure 2) Agent a_i can* communicate indirectly *with agent a_j iff $\exists\ \{a_k \in \mathcal{A}, i \leq k < j\}$ and $\exists\ \{\tau_k within \mathring{p},\ i \leq k < j\}$ such as $a_{k+1} \in$ Vicinity(a_k, τ_k).*

In case of an indirect communication, a_i and a_j may communicate through several agents forming a *daisy chain*. As Vicinity is symmetric but not transitive, direct communication is symmetric whereas indirect communication is oriented from an agent to another one. Each communication from a_i to a_j is associated with a couple $(\tau_i, \tau_j) \in \mathbb{T}^2$ with τ_i the emitting date of a_i and τ_j the receipt date of a_j. We will write: a_i communicates with a_j at (τ_i, τ_j). In case of a direct communication, $\tau_i = \tau_j$.

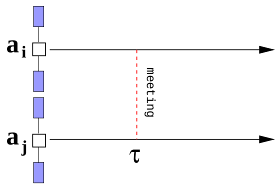

Fig. 1. Direct communication

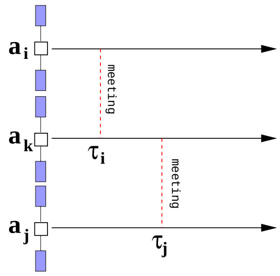

Fig. 2. Indirect communication

The constellation (agents, clock and Vicinity) is knowledge that all the agents hold in common.

2.2 Private knowledge

The private knowledge of an agent within the constellation is defined as follows:

Definition 4 (Knowledge) *A* piece of knowledge *$K^{\tau}_{a_i}$ of agent a_i at time τ is a triplet $\langle D_{K^{\tau}_{a_i}}, A_{K^{\tau}_{a_i}}, \tau_{K^{\tau}_{a_i}} \rangle$:*

- *$D_{K^{\tau}_{a_i}}$ is a task t_j or an intention $I^{a_k}_{t_j}$ of a_k about t_j, $a_k \in \mathcal{A}$;*
- *$A_{K^{\tau}_{a_i}} \subseteq \mathcal{A}$ is the subset of agents knowing $K^{\tau}_{a_i}$;*
- *$\tau_{K^{\tau}_{a_i}} \in \mathbb{T}$ is the date when $D_{K^{\tau}_{a_i}}$ was created or updated.*

Let $\mathcal{K}_{a_i}^{\tau}$ be the knowledge of agent a_i at time τ: $\mathcal{K}_{a_i}^{\tau}$ is the set of all the pieces of knowledge $K_{a_i}^{\tau}$.

From $\mathcal{K}_{a_i}^{\tau}$, we define $\mathcal{T}_{a_i}^{\tau} = \{t_1 \dots t_m\}$ the set of tasks known by agent a_i at time τ ; and $\mathcal{I}_{a_i}^{\tau} = (I_{t_j}^{a_k})$ the matrix of the intentions known by agent a_i at time τ. Each agent a_i has resources available to realize only a subset of $\mathcal{T}_{a_i}^{\tau}$.

2.3 Tasks

Each agent within the constellation knows some *tasks* to realize.

Definition 5 (Task) *A* task t *is an observation request associated with a priority*[3] $prio(t_j) \in \mathbb{N}^*$ *and with a boolean b_{t_j} that indicates whether t_j has been realized or not.*

The tasks may be constrained in two ways:

– **mutual exclusion**: it is an agent's constraint meaning that it cannot realize several tasks at the same time τ ;
– **composition** of n tasks: all the n tasks must be realized, it is useless to realize only a strict subset of them. Formally,

  Definition 6 (Compound task) *A* compound task *is a subset $\mathcal{T}$ of tasks such as* $(\exists t_i \in \mathcal{T}, t_i \text{ is realized}) \Rightarrow (\forall t_j \in \mathcal{T}, t_j \neq t_i \text{ must be realized})$.

Moreover when a task is realized by an agent, it is redundant if it has already been realized by another agent:

Definition 7 (Redundancy) *Let a_i be an agent that realizes a task t_k at time $\tau \in \mathbb{T}$. There is a* redundancy *about t_k if and only if $\exists\ a_j \in \mathcal{A}$ and $\exists\ \tau' \in \mathbb{T}$ ($\tau' \leq \tau$) such as a_j has realized t_k at time τ'.*

2.4 Intentions

An intention represents an agent's attitude towards a given task.

Definition 8 (Intention) *Let $I_{t_j}^{a_i}$ be the* intention *of agent a_i towards task t_j. $I_{t_j}^{a_i}$ is a modality of proposition (a_i* **realizes** *t_j) :*

– $\Box$ *(*commitment*): a_i is committed to realize t_j*
– $\Diamond$ *(*proposal*): a_i proposes to realize t_j*
– $\Box\neg$ *(*strong withdrawal*): a_i will not realize t_j*
– $\Diamond\neg$ *(*weak withdrawal*): a_i does not propose to realize t_j*

A realization date $rea(I_{t_j}^{a_i}) \in \mathbb{T} \cup \{\emptyset\}$ and a download date $tel(I_{t_j}^{a_i}) \in \mathbb{T} \cup \{\emptyset\}$ are associated with each intention.

[3] In the space domain, 1 stands for the highest priority whereas 5 is the lowest. Consequently the lower $prio(t_j)$, the more important task t_j.

The set of an agent's intentions corresponds to its current plan. Each commitment or proposal means that the associated task is planned. The tasks associated with withdrawals are not planned. We assume that each agent has an individual planner. Planning is a three-step process:

1. From the set of unrealized tasks known by a_i at time τ, a_i computes an optimal local plan under two criteria[4]:
   – maximize the number of planned tasks;
   – minimize the number of unplanned high priority tasks.
2. The intentions of agent a_i about tasks t_j at time $(\tau - 1)$ constrain the planning process (step 1):
   – tasks associated with a commitment ($\Box$) are *always* planned;
   – tasks associated with a strong withdrawal ($\Box\neg$) are *never* planned.
3. Agent a_i's plan at time τ modifies its intentions as follows:
   – each new planned task generates a proposal ($\Diamond$);
   – each new unplanned task is set aside ($\Diamond\neg$).

We can notice that the commitments ($\Box$) and strong withdrawals ($\Box\neg$) are not generated by the planning process. We will see in Section 5 that these intentions are generated by a collaboration process between the agents.

2.5 Trust in proposals

An agent that receives a given proposal at time τ cannot be sure that this intention will be the same at time τ' ($\tau' > \tau$). Indeed as the environment is dynamic, an agent may receive new tasks or new intentions and modify its plan, i.e. its own proposals, accordingly. The more time between the generation of a given proposal and the realization date, the less an agent can trust it. However a further confirmation transmitted by the agent that has generated this proposal increases the associated trust again. This mechanism is described in more details in [2]. Here, we define formally the last confirmation of a proposal:

Definition 9 (Last confirmation) *Let a_i be an agent, $I_{t_j}^{a_j}$ a proposal of an agent a_j about a task t_j known by a_i. The* last confirmation *of proposal $I_{t_j}^{a_j}$ for a_i at time τ is:*

$$\tau^* = \max_{\tau_{K_{a_i}^{\tau}} < \tau_j, \tau_i < \tau} \{\tau_j : a_j \textit{ communicates with } a_i \textit{ at } (\tau_j, \tau_i)\}$$

As the agents do not have a model of the environment, they cannot predict the arrival of new tasks. However as time passes, an agent meets other agents and each meeting is an opportunity to receive new tasks and revise its intentions. Consequently an agent's trust about a given proposal is defined from the number of meetings between the last confirmation and the realization date. This number is based on Vicinity therefore each agent can compute its own trust in the others' proposals.

[4] The individual planning process itself is beyond the scope of our work. The mono-agent planning problem may be adressed with many techniques such as constraint programming or HTN planning.

Definition 10 (Meetings) *Let a_i be an agent, $I_{t_j}^{a_j}$ a proposal known by a_i and τ the current date. Let τ^* be the last confirmation of $I_{t_j}^{a_j}$ for a_i at time τ. The number of agents $M_{\tau^*}^{a_i}(I_{t_j}^{a_j})$ agent a_j will meet between τ^* and $rea(I_{t_j}^{a_j})$ is* $M_{\tau^*}^{a_i}(I_{t_j}^{a_j}) = |\bigcup_{\tau^*<\tau'<rea(I_{t_j}^{a_j})} Vicinity(a_j, \tau')|$

Definition 11 (Trust) *Let a_i be an agent, $I_{t_j}^{a_j}$ a proposal known by a_i and τ the current date. a_i* trusts *a_j about $I_{t_j}^{a_j}$ if and only if $M_{\tau^*}^{a_i}(I_{t_j}^{a_j}) = 0$.*

We can notice that the trust criterion is hard: an agent is not trusted if it meets another agent before realizing its proposal ($M_{\tau^*}^{a_i}(I_{t_j}^{a_k}) = 0$). This pessimistic assumption can be relaxed (e.g. $M_{\tau^*}^{a_i}(I_{t_j}^{a_k}) \leq 1$).

3 Communication

The agents have to reason on a common knowledge in terms of tasks and intentions. A communication protocol is defined to allow an agent to know what the other agents know. Because of the communication delays, this common knowledge concerns only a subset of agents.

3.1 An epidemic protocol

An epidemic protocol based on overhearing [11] has been proposed [3]. The agents use every communication opportunity even to communicate information that does not concern themselves:

1. each agent a_i considers its own knowledge changes;
2. a_i communicates the changes to $a_j \in \text{Vicinity}(a_i, \tau)$;
3. a_j updates its own knowledge thanks to the timestamp $\tau_{K_{a_i}^{\tau}}$.

It has been proved that, in a set of n agents where a single agent knows a piece of information, an epidemic protocol needs $\mathcal{O}(\log n)$ communication rounds to completely propagate this information [12]. During a communication round, each agent executes a communication step that has a polynomial complexity in the number of agents and tasks [3].

3.2 Common knowledge

Thanks to this communication protocol, we define the notion of common knowledge in terms of intentions:

Definition 12 (Common knowledge) *At time τ, agent a_i knows that agent a_j knows the intention $I_{t_j}^{a_i}$ captured by $K_{a_i}^{\tau}$ iff:*

- *$a_j \in A_{K_{a_i}^{\tau}}$* ***or***
- *a_i communicated with a_j at (τ_i, τ_j) such as $\tau_{K_{a_i}^{\tau}} \leq \tau_i$, $\tau_j \leq \tau$.*

4 Coalitions

4.1 State-of-the-art

A coalition is an agent organization with a short life cycle. It is formed in order to realize a given goal and is destroyed when the goal is achieved. Through a coalition, each agent tries to maximize its personal outcome. In the literature, the methods dedicated to coalition formation are based on the exploration of the lattice of the possible coalition structures [15, 20]. In order to find the optimal structure, the agents often have uncertain and (or) incomplete information on the other agents' costs and preferences: they need to use heuristics [10] or trust [16] to evaluate a coalition value.

Generally speaking, these methods have two limits.
On the one hand, they are often centralized, they assume that all tasks are known by all agents and they are performed off-line [7, 8, 13, 17] ; or they use an auctioneer (or other kinds of hierarchy) [1, 18] that centralizes the information and organizes the negotiations.
As far as communications are concerned, methods based on the system organization structure consider constrained communications: agents can communicate through a hierarchy [1, 18] or in a vicinity [19]. These constraints are associated with a communication cost [21]. However in a real dynamic environment, agents are not always able to exchange information and may have to decide alone. Moreover, some tasks cannot wait for the complete computation of the coalition structure and must be realized quickly. Thus, these methods are very sensitive to the system dynamics.

Be that as it may, the coalition formation mechanisms are interesting for three reasons: (1) agents gather in order to realize a collective task; (2) the short life cycle of coalitions is adapted to dynamic environments; (3) agents search for efficient solutions under uncertain and (or) incomplete information.
In our application, compound tasks require that some agents should realize some subsets of tasks jointly. However these joint realizations cannot be planned by the agents' individual planners as an agent does not plan for the others. In order to dynamically organize the agents, we will consider a decentralized coalition formation mechanism taking into account the features of our problem, i.e. cooperative agents and constrained communications. The mechanism is as follows:

1. Agents build maximal-size coalitions with respect to their own knowledge;
2. These coalitions are refined as the agents meet to remove useless agents.

4.2 Definitions

Coalitions are defined as follows:

Definition 13 (Coalition) *A* coalition *C is a triplet $\langle A, O, P \rangle$:*
- *$A \subseteq \mathcal{A}$ is a subset of agents that are the* members *of the coalition;*

– *O is the set of tasks that are the* goals *of the coalition;*
– *P is the set of tasks that are in the* power *of the coalition.*

A coalition C can be in different states:
– *C is* complete *iff $O \subseteq P$;*
– *C is* minimal *iff C is complete and A is minimal for inclusion ($\subseteq$).*

The next section will show how coalitions, which are built and managed *locally* by each agent, allow agents to collaborate.

5 Collaboration *via* coalitions

Coalitions are built and managed locally by each agent, given the knowledge it has about the other agents through communication. Indeed each agent uses the coalition notion to reason and adapt its own intentions to the others' intentions. Therefore, coalitions are formed implicitly through intentions but are not explicitly built by the multi-agent system. Each agent:

1. computes the current coalition structure according to its point of view;
2. checks whether it should join a coalition to increase its power;
3. checks whether it should withdraw from a coalition to minimize it ;
4. modifies its intentions accordingly.

5.1 Computation of the coalition structure

Each agent a_i generates the current coalition structure as follows:

1. a_i organizes the set of tasks $\mathcal{T}_{a_i}^{\tau}$ as a partition $\{\mathcal{T}_1 \dots \mathcal{T}_h\}$ according to the compound tasks;
   Example 1 *Let $\mathcal{T}_{a_i}^{\tau}$ be $\{t_1, t_2, t_3, t_4, t_5\}$. Let us suppose that tasks t_1 and t_2 form a compound task as well as t_4 and t_5. Then $\mathcal{T}_{a_i}^{\tau}$ is organized as $\{\{t_1, t_2\}$, $\{t_3\}$, $\{t_4, t_5\}\}$.*
2. each $\mathcal{T}_i$ is the goal of a single potential coalition; as subsets $\mathcal{T}_i$ are disjoint[5], the number of potential coalitions generated by agent a_i is given by the number of compound tasks a_i knows;
3. from agent a_i's point of view, the potential coalition members for subset $\mathcal{T}_i$ are defined as: $\{a_k \in \mathcal{A} : \exists\, t_j \in \mathcal{T}_i \,/\, \exists\, I_{t_j}^{a_k} \in \mathcal{K}_{a_i}^{\tau}$ such that $I_{t_j}^{a_k} \in \{\Box, \Diamond\}\}$
   Example 2 *Let us resume Example 1. Let us consider t_3 and suppose that $I_{t_3}^{a_i} = \Diamond$ and $I_{t_3}^{a_k} = \Box$. a_i can build coalition $C = \langle\{a_i, a_k\}, \{t_3\}, \{t_3\}\rangle$. This coalition is complete but not minimal because $\{a_i, a_k\}$ is not minimal for inclusion. Notice that a_i plans t_3 even if it knows that a_k did the same. Indeed the others' intentions are not taken into account in the planning step: they will be taken into account in the collaboration steps (2, 3, 4).*
4. then the power of each potential coalition is defined as: $P = \{t_j \in O | \exists a_i \in A : I_{t_j}^{a_i} \in \{\Box, \Diamond\}\}$

A potential coalition may be minimal (thus complete), complete and not minimal or incomplete.

[5] The compound tasks are assumed disjoint but notice that they can overlap without modifying the collaboration process.

5.2 An incentive to join coalitions

An incomplete coalition means that at least one goal task is not within the coalition power. But the more tasks within the coalition power, the more goal tasks become important because a coalition must realize all its goal tasks. If the coalition remains incomplete, all its members waste their resources.

When agent a_i computes the current coalition structure according to its knowledge, it can detect incomplete coalitions. As a_i is cooperative, it should be incited to modify its intentions and complete these coalitions when planning. In order to do that, we propose to increase the priorities of the goal tasks of the incomplete coalitions. In the following, we will note $prio(t)'$ the priority of task t a_i uses for its next planning step. Notice that $prio(t)'$ is a local priority only used by a_i. The commercial or physical priority $prio(t)$ of task t remains the same.

Protocol 1 (Join a coalition) *For each incomplete coalition $C = \langle A, O, P \rangle$, agent a_i computes: $\forall\ t \in O$, $prio(t)' \leftarrow \frac{prio(t)}{1+|P|}$.*

The agent is encouraged to join a coalition if and only if the goal of the coalition is to realize a compound task that is partially planned.

Remarks: as far as singletons $\{t_j\}$ are concerned,

- if t_j is not planned by a_i, it is because it does not satisfy the optimization criteria (Section 2.4); therefore a_i does not build any coalition concerning t_j and the priority of t_j remains the same;
- if t_j is planned, the coalition concerning t_j is complete and its priority remains the same.

Example 3 *Let us resume Example 1. Let us consider $\{t_1, t_2\}$ and suppose that $I_{t_1}^{a_i} = \Diamond\neg$, $I_{t_2}^{a_i} = \Diamond\neg$, $I_{t_1}^{a_k} = \Diamond\neg$ and $I_{t_2}^{a_k} = \Box$. a_i can build coalition $C = \langle\{a_k\}, \{t_1, t_2\}, \{t_2\}\rangle$. This coalition is incomplete. So a_i applies Protocol 1. As a_k is already a member of the coalition, the priorities of t_1 and t_2 are halved for a_i. Therefore at its next planning step, a_i is more likely to plan t_1 or t_2 instead of other tasks.*

This mechanism is *stable*, i.e. two successive incentive steps are consistent. For instance, an agent is not encouraged to give up a given task in order to realize another one, then *ceteris paribus* is not encouraged to give up the latter to realize the former.

5.3 Minimizing coalitions: conflicts

A complete and non minimal coalition has the power to realize its goals with useless agents, i.e. agents that have redundant intentions. Within a coalition, an agent has to consider the agents that have planned the same tasks as it has, then to make a decision about modifying or not its own intentions. There is a conflict between two agents within a coalition if they have planned the same task(s). Formally:

Definition 14 (Conflict) *Let a_i, a_j be two agents and C a coalition $\langle A, O, P\rangle$ such as $\{a_i, a_j\} \subseteq A$. There is a* conflict *between a_i and a_j iff $\exists\ t \in P$ such as $I_t^{a_i} \in \{\Box, \Diamond\}$ and $I_t^{a_j} \in \{\Box, \Diamond\}$. It is a* ***soft conflict*** *iff either a_i communicates with a_j at (τ_i, τ_j) such as $\tau_{I_{t_j}^{a_i}} < \tau_i$ and $\tau_j < \min(rea(I_{t_j}^{a_i}), rea(I_{t_j}^{a_j}))$ or a_j knows agent a_i's intention about t_j. Else it is a* ***hard conflict****.*

Example 4 *Let us resume Example 2.The coalition is not minimal: there is a conflict about task t_3 between agents a_i and a_k. So a_i has to make a decision in order to withdraw ($\Box\neg$), to keep its intention ($\Diamond$) or to commit ($\Box$).*

Proposition 1 (Symmetry) *Let a_i be an agent and A^* the set of agents with which it is in conflict about task t_j. $\forall\ a_j \in A^+$, the conflict is* symmetric. *$\forall\ a_j \in A^-$, the conflict is* asymmetric.

Proof 1 *Let a_i be an agent and A^* the set of agents with which it is in conflict about task t_j.*

- *$\forall\ a_j \in A^+$, a_i knows $I_{t_j}^{a_j}$. Conversely either a_j knows $I_{t_j}^{a_i}$, or $\exists\ \tau_i$, $\tau_j \in \mathbb{T}$ such as a_i communicates with a_j at (τ_i, τ_j) and such that $\tau_{I_{t_j}^{a_i}} < \tau_i$ and τ_j $<$ min $(rea(I_{t_j}^{a_i})$, $rea(I_{t_j}^{a_j}))$. In both cases, the conflict is symmetric and it is a soft conflict.*
- *$\forall\ a_j \in A^-$, a_j does not know $I_{t_j}^{a_i}$ and it will not know it before the date* min *$(rea(I_{t_j}^{a_i}), rea(I_{t_j}^{a_j}))$. So a_j is not and will not be aware of the conflict ; it is a hard conflict.*

Both soft and hard conflicts are dealt with through protocols based on the agents' expertise for realizing the task.

5.4 Minimizing coalitions: the expertise criterion

As we are seeking to optimize the system reactivity, it is better that the agents realize the tasks as soon as possible and use the fewest resources possible[6]. Let us aggregate both criteria in a single expertise criterion. Formally:

Definition 15 (Expertise) *Let $A^* \subseteq \mathcal{A}$ be a set of agents in conflict about a task t_j. Let us note $rea^* = \min\limits_{a_i \in A^*} rea(I_{t_j}^{a_i})$ the earliest realization date for task t_j. The* expert *agent for t_j is defined using the following distance:*

$$a^* = \arg\min_{a_i \in A^*} ||(rea(I_{t_j}^{a_i}) - rea^*, tel(I_{t_j}^{a_i}) - rea^*)||$$

Figure 3 is a representation of the expertise criterion for a task t in the plan $(rea(I_t^{a_i}),\ tel(I_t^{a_i}))$, $a_i \in A^*$. The origin rea^* is the earliest realization date for t and intention (rea^*, rea^*) is the ideal intention corresponding to an agent being able to realize t at time rea^* and download the corresponding picture immediately. tel^* is the latest download date for t, if t is realized at time rea^*. Obviously $tel(I_t^{a_i}) > rea(I_t^{a_i})$ therefore only the hatched part is meaningful.

[6] Using fewer resources means keeping the pictures in the satellite memory for the shortest time possible, i.e. downloading them as soon as possible.

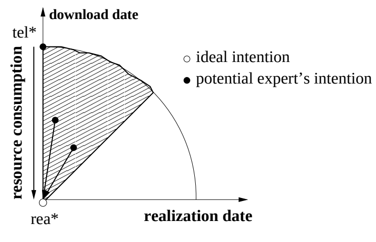

Fig. 3. Expertise criterion for task t

Any point within the hatched part is a potential intention $I_t^{a_i}$ about t. The resource consumption, i.e. how long the picture corresponding to t will remain in the memory of the satellite, is defined as a duration. The distance between a potential intention and rea^* represents the projection of the time criteria on the plan $(rea(I_t^{a_i}), tel(I_t^{a_i}))$. The expert agent for t is the one that minimizes this distance.

5.5 Minimizing coalitions: coordination strategies

In order to solve a conflict, three strategies are defined. (1) With a *secure strategy* a_i maintains its proposal ($\Diamond$) if it does not trust the other agents about their intentions; as these agents are likely to change their intentions, this strategy maintains redundancies to make sure that the task will be realized. (2) With a *collaboration strategy* a_i commits ($\Box$) if it is the expert agent, therefore deciding on a part of the current coalition structure. (3) With an *opportunistic strategy* a_i strongly withdraws ($\Box\neg$) if the expert agent is trusted, therefore minimizing the size of the coalition and saving resources for further tasks.

From the three strategies, two conflict solving protocols are defined:

Protocol 2 (Hard conflict) *Let A^* be the set of the coalition members with which agent a_i is in conflict about task t_j such that $A^- \neq \emptyset$. a_i is aware of the conflict and applies:*

1. `if` $\min_{a_k \in A^-} M_{\tau^*}^{a_i}(I_{t_j}^{a_k}) > 0$ `then` $I_{t_j}^{a_i} \leftarrow \Diamond$
2. `else` $I_{t_j}^{a_i} \leftarrow \Box\neg$

In case of a hard conflict, the agent that is aware of the conflict applies (1) the secure strategy if it does not trust the agents within the conflict ; else (2) if it trusts them, the aware agent applies the opportunistic strategy.

Protocol 3 (Soft conflict) *Let A^* be the set of the coalition members with which agent a_i is in conflict about task t_j such that $A^+ \neq \emptyset$. Let rea^* be $\min_{a_k \in A^+} rea(I_{t_j}^{a_k})$. Then agent a_i applies:*

1. `if` $a_i = \arg\min_{a_k \in A_+} ||(rea(I_{t_j}^{a_k}) - rea^*, tel(I_{t_j}^{a_k}) - rea^*)||$ `then` $I_{t_j}^{a_i} \leftarrow \Box$
2. `else` *let a^* be the expert agent:*
   (a) `if` $M_{\tau^*}^{a_i}(I_{t_j}^{a^*}) > 0$ `then` $I_{t_j}^{a_i} \leftarrow \Diamond$
   (b) `else` $I_{t_j}^{a_i} \leftarrow \Box\neg$

For soft conflicts, each agent computes the expert agent. (1) If it is the expert agent, it commits. (2.a) If not, it applies the secure strategy if it does not trust the expert (2.b) If it trusts the expert, it applies the opportunistic strategy.

6 Experiments

The different mechanisms and protocols we have described have been implemented. Two metrics are considered to compare the results: the number of realized tasks and the number of realized tasks without redundancy. The first metric corresponds to the number of distinct singleton or compound tasks realized. Experiments have been conducted on three kinds of constellations:

– *isolated*: no communication;
– *informed*: agents communicate only about tasks and coordinate *a posteriori* by withdrawing already realized tasks from their plans;
– *coordinated*: agents communicate about tasks and intentions and coordinate *a priori* thanks to coalition formation.

6.1 Reference framework : static simulations

The reference experiments are based on a scenario with 3 agents and 100 tasks. It is a static scenario, meaning that the initial set of tasks is fixed and new tasks will not appear during the simulations. Four parameters are considered: the task density, the task composition rate, the quantity of the agents' memory resources and the number of hours needed to complete the tasks. For each parameter value, we have launched 100 simulations and computed the average result.

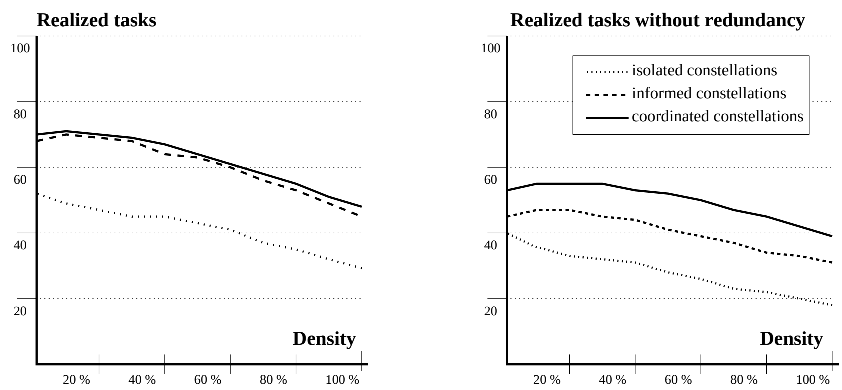

Fig. 4. Realized tasks (with and without redundancy) under density constraint

Definition 16 (Density) *The task density represents how close to each other the tasks are. The closer the tasks, the more they are likely to be in mutual exclusion.*

(Figure 4) The results for informed and coordinated constellations are better than for isolated constellations. Although informed and coordinated constellations realize nearly the same number of tasks (with a slight advantage for coordinated constellations), coordination allows the number of minimal (i.e. optimal) coalitions to be increased drastically. However we can notice that the difference between informed and coordinated constellation in terms of realized tasks is not so important: this comes from the fact that these experiments are within a static world, new tasks do not appear during the simulations: when resources are saved by an agent, they are not necessarily reallocated. In a dynamic world with new tasks and no bounded temporal horizon, resources will be reallocated.

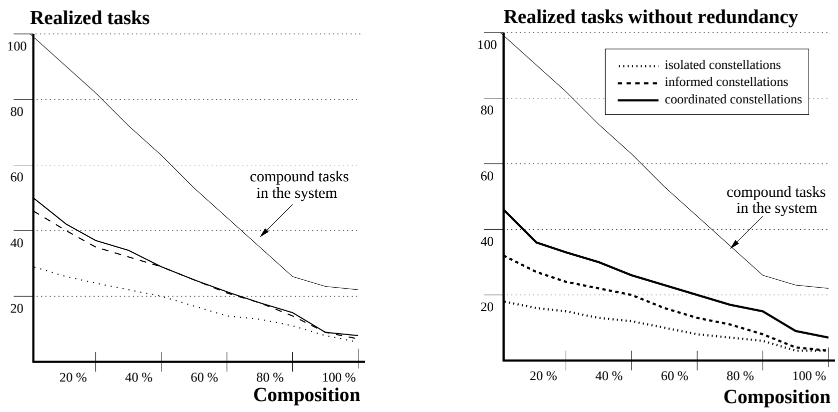

Fig. 5. Realized tasks (with and without redundancy) under composition constraint

Definition 17 (Composition) *The task composition represents the percentage of tasks that are in mutual exclusion with another task and that are jointly the goal of a potential coalition.*

(Figure 5) We can notice that increasing the composition ratio decreases the number of potential coalitions, and consequently the maximal number of complete and minimal coalitions. This affects the informed and coordinated constellations more than the isolated ones: the relative loss of efficiency in terms of complete and minimal coalitions is higher. However, the absolute results for informed and coordinated constellations are better than for the isolated ones.

From the initial scenario, we have run simulations where agents must realize all tasks. Our metric is now the number of hours that are needed to complete all the tasks. Figure 6 shows results with the density constraint and Figure 7 shows results with the resource constraint. In both cases, we can notice that informed and coordinated constellations outrun the isolated constellations. The benefits in swiftness range from 30% to 70% according to the task density; and

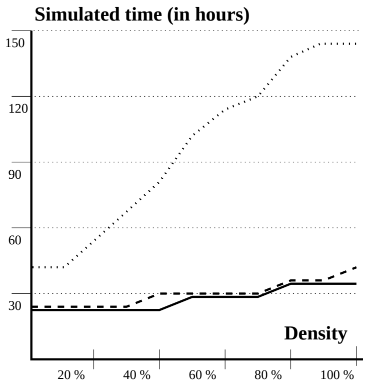

Fig. 6. Task density

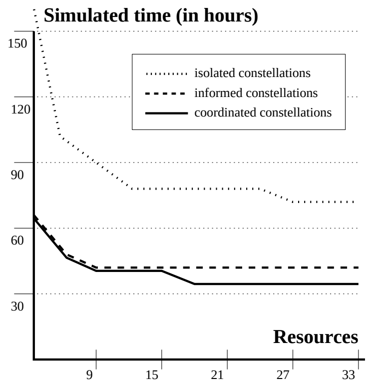

Fig. 7. Resource limitation

from 60% to 40% according to the resource limitation. Coordination allows to sligthly increase this benefit (a gain close to 6 hours of reactivity).

6.2 Real-world framework : dynamic simulations

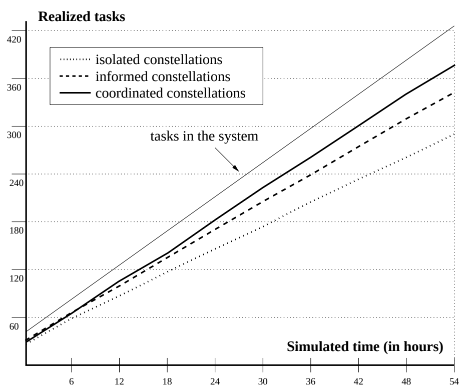

Fig. 8. Tasks

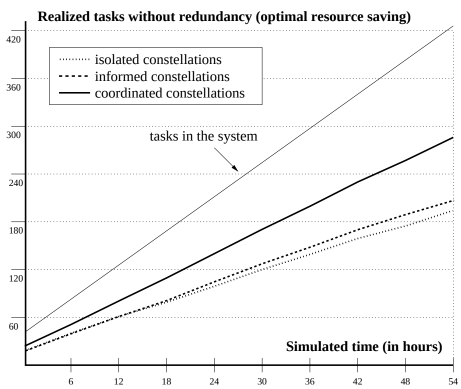

Fig. 9. Tasks with no redundancy

The next experiments are based on a dynamic scenario with 3 agents. Every 6th hour, the ground stations send 40 new compound tasks (including at least 2 singleton tasks) to the agents. We have launched 25 simulations and computed the average result. Two metrics are considered: the number of realized tasks (Figure 8) and the number of realized tasks without redundancy (Figure 9).

As previously informed and coordinated constellations outperform isolated ones. However we can notice that the benefits increase as time passes. Indeed incremental coordination allows coordinated constellations to realize more tasks than the other kinds of constellations. And as time passes the difference between informed and coordinated constellations increases: incremental coordination allows coordinated constellations to efficiently save and reallocate resources.

7 Conclusion

We have proposed a collaboration method for physical agents that communicate from time to time in a dynamic environment. This method has been applied to a constellation of satellites. A communication protocol has been proposed in order to build mutual knowledge (in terms of tasks and intentions) as the agents meet.

The collaboration process is an online incremental coalition formation that proceeds through a *planning - communication - collaboration* loop within each agent. Each agent builds an initial plan. From its knowledge, each agent builds the potential coalitions that can realize the tasks it knows. Afterwards these coalitions are refined thanks both to an *incentive* mechanism and an *optimization* mechanism. The agents' communication capabilities on the one hand and conflict definitions on the other hand allow us to define protocols that refine the coalition structure dynamically and adapt it to new knowledge.

The experimental results are promising. In a static world (i.e. bounded temporal horizon, bounded initial set of tasks, no new task) the coalition formation mechanism allows the resource consumption to be minimized; nevertheless this does not necessarily have an impact on the number of realized tasks. However in a dynamic world (i.e. new tasks and unbounded temporal horizon), the saved resources are reallocated in a incremental way and the number of realized tasks is increased.

Future work will deal with the possible failures of the agents and the consequences on the other agents' trusts. The communication complexities of informed (linear in the number of tasks and agents) and coordinated (polynomial in the number of tasks and agents) constellations versus the number of saved and reallocated resources are also worth studying. Furthermore simulations involving a higher number of satellite agents (up to 20) will be performed to verify scalability of our approach.

Acknowledgements

We would like to thank Marie-Claire Charmeau (CNES - The French Space Agency) and Serge Rainjonneau (Thales Alenia Space) for their relevant remarks on this work.

References

1. S. Abdallah and V. Lesser. Organization-based cooperative coalition formation. In Proceedings of the IEEE IAT'04 (2004) 162–168

2. G. Bonnet and C. Tessier. A Trust Model Based On Communication Capabilities For Physical Agents. In Proceedings of the TRUST workshop at the 7th AAMAS (2008)
3. G. Bonnet and C. Tessier. Collaboration among a satellite swarm. In Proceedings of the 6th AAMAS (2007) 275–282
4. C. Brooks and E. Durfee. Congregation formation in multiagent systems. Journal of Autonomous Agents and Multiagent Systems (2003) Vol. 7:145–170
5. S. Damiani and G. Verfaillie and M.-C. Charmeau. An Earth watching satellite constellation: how to manage a team of watching agents with limited communications. Proceedings of the 4th AAMAS (2005) 455–462
6. F. Dignum. Autonomous agents with norms. Artificial Intelligence and Law (1999) Vol. 7:69–79
7. V. Dung Dang, R.K. Dash, A. Rogers and N.R. Jennings. Overlapping coalition formation for efficient data fusion in multi-sensor networks. In Proceedings of the 21st AAAI (2006) 635–640
8. H.J. Goradia and J.M. Vidal. An equal excess negotiation algorithm for coalition formation. In Proceedings of the 6th AAMAS (2007) 1052–1054
9. B. Horling and V. Lesser. A survey of multi-agent organizational paradigms. In The Knowledge Engineering Review (2004) Vol. 19:281–316
10. S. Kraus, O. Shehory and G. Taase. Coalition formation with uncertain heterogeneous information. In Proceedings of the 2nd AAMAS (2003) 1–8
11. F. Legras and C. Tessier. LOTTO: group formation by overhearing in large teams. In Proceedings of 2nd AAMAS (2003)
12. B. Pittel. On spreading a rumor. SIAM Journal of Applied Mathematics (1987) Vol. 47:213–223
13. T. Rahwan, S.D. Ramchurn, V. Dung Dang and N.R. Jennings. Near-optimal anytime coalition structure generation. In Proceedings of the 21st IJCAI (2007) 2365–2371
14. T. Sandholm. Contract types for satisficing task allocation. In Proceedings of the AAAI Spring Symposium on Satisficing Models (1998) 23–25
15. T. Sandholm, K. Larson, M. Andersson, O. Shehory and F. Tohm. Coalition structure generation with worst case guarantees. Artificial Intelligence (1999) Vol. 111(1-2):209–238
16. S. Sen, I. Goswami and S. Airiau. Expertise and trust-based formation of effective coalitions: an evaluation of the ART testbed. In Proceedings of the ALAMAS workshop at the 5th AAMAS (2006)
17. O. Shehory and S. Kraus. Feasible formation of coalitions among autonomous agents in non-super-additive environments. Computational Intelligence (1999) Vol. 15(3):218–251
18. M. Sims, C. Goldman and V. Lesser. Self-organization through bottom-up coalition formation. In Proceedings of the 2nd AAMAS (2003)
19. L.-K. Soh and C. Tsatsoulis. Utility-based multiagent coalition formation with incomplete information and time constraints. In Proceedings of the IEEE International Conference on SMC (2003)
20. D. Thanh-Tung, B. Frankovic, C. Sheahan and I. Bundiska. Using agent coalitions for improving plan quality. Intelligent Systems at the Service of Mankind (2005) Vol. 2:351–364
21. F. Tohmé and T. Sandholm. Coalition formation processes with belief revision among bounded-rational self-interested agents. Journal of Logic and Computation (1999) Vol. 9(6):793–815