An 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 incremental coalition formation in order to optimize individual plans so as to satisfy collective objectives. This involves a communication protocol, common knowledge 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 [5], contracts [17] or organizations [10] are considered. As far as *physical agents* such as robots or satellites are concerned, information sharing and coordination depend on communication constraints. 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. networks of satellites) allow to consider joint activities and ensure functional robustness [4]. We consider a set of 3 to 16 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, geographically close requests cannot be realized by the same satellite. Likewise each satellite has limited memory resources and can realize only a given number of requests before downloading. Notice that in the space lexicon downloading means transferring data to a ground station (i.e. the pictures taken when a request is realized). 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.

Consequently the problem we focus on is a distributed 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 (refer to Definition 5) 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.

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 us from having a centralized view of the problem and therefore a centralized resolution. In the literature, two distributed approaches are considered to control a satellite constellation:

1. the hierarchical approach [2, 7, 8, 20] where a leading satellite plans for the others: this approach is very sensitive to local failures and to the arrival of new tasks;
2. the decentralized approach [4] but ISL are not considered to increase the quality of the local plans.

As new tasks arrive as time goes by, a decentralized approach in which ISL are taken into account must be considered. 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 common knowledge;
3. agents build and revise coalitions that influence their individual plans.

This paper is organized as follows. In Section 2 we will describe how agents are modelled in a multi-agent system. In Section 3 we will present how agents communicate and reason to build a trusted common knowledge. The organization model is presented in Section 4 and the formal mechanism is described in Section 5. Before concluding Section 6 will show results about performance and scalability of the approach.

2 The agents

2.1 The multi-agent system structure

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

Definition 1 (Constellation) *The* constellation $\mathcal{S}$ *is a triplet* $\langle \mathcal{A}, \mathbb{T}, Vicinity \rangle$ *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* $Vicinity : \mathcal{A} \times \mathbb{T} \mapsto 2^{\mathcal{A}}$ *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.*

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* $a_i \in \mathcal{A}$*. The Vicinity period* $P \in \mathbb{T}$ *is the* lowest common multiple *of set* $\{p_1 \dots p_n\}$.

In the remainder, we will note $\mathbb{T}_P \subset \mathbb{T}$ the time interval of duration P such that $\mathbb{T}_P = [0 \dots P]$.

The constellation (agents, clock and Vicinity) is knowledge that all the agents hold in common. Nonetheless each agent also holds private knowledge.

2.2 Observation requests modelled as tasks

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

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

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

The tasks may be constrained in two ways: (1) *mutual exclusion* meaning that a given agent cannot realize several tasks at the same time τ ; (2) *composition* of n tasks meaning that all the n tasks must be realized : it is useless to realize only a strict subset of them. Formally,

Definition 4 (Compound task) *A* compound task *is a subset* $\mathcal{T}$ *of tasks such that* $(\exists t_i \in \mathcal{T}, t_i \text{ is realized}) \Rightarrow (\forall t_j \in \mathcal{T}, t_j \neq t_i, t_j \text{ must be realized})$.

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

Definition 5 (Redundancy) *Let* a_i *be an agent that realizes a task* t *at time* $\tau \in \mathbb{T}$*. There is a* redundancy *about* t *if and only if* $\exists\ a_j \in \mathcal{A}$ *and* $\exists\ \tau' \in \mathbb{T}$ *(*$\tau' \leq \tau$*) such that* a_j *has realized* t *at time* τ'.

2.3 Agents' attitudes modelled as intentions

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

Definition 6 (Intention) *Let $I_t^{a_i}$ be the* intention *of agent a_i towards task t. $I_t^{a_i}$ is a modality of proposition (a_i* ***realizes*** *t) :*

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

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

2.4 Agents' private knowledge

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

Definition 7 (Knowledge) *A* piece of knowledge *$K_{a_i}^\tau$ of agent a_i at time τ is a triplet $\langle D_{K_{a_i}^\tau}, A_{K_{a_i}^\tau}, \tau_{K_{a_i}^\tau} \rangle$:*

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

2.5 The individual planning process

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. Notice that the individual planning process itself is beyond the scope of our work. Consequently 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:
   - maximize the number of planned tasks;
   - minimize the number of unplanned high priority tasks.
2. The intentions of agent a_i about the tasks t 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 generates a weak withdrawal ($\Diamond\neg$).

We can notice that 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.

3 Building a trusted common knowledge

The agents have to reason on common knowledge about tasks and intentions. Consequently a communication protocol is defined to allow an agent to know what the other agents know. Communication is based on Vicinity: when two agents meet they can communicate. Consequently the Vicinity structure influences the communication capabilities.

3.1 Communication

We define communication within the constellation as follows:

Definition 8 (Communication) *Let $\mathcal{S}$ be a constellation and a_i, $a_j \in \mathcal{A}$. An agent a_i can communicate with an agent a_j in two ways:*

- directly *iff* $\exists\ \tau_i \in \mathbb{T}_P$ *such that* $a_j \in$ *Vicinity*(a_i, τ_i)*;*
- indirectly *iff* $\exists\ l \in \mathbb{N}^*$ *such that* $\exists\ \{(a_{\tau_k}, \tau_k) \in \mathcal{A} \times \mathbb{T}, k \in [0 \dots l]\}$ *where:*
  1. $a_{\tau_0} \in$ *Vicinity*(a_i, τ_i)*;*
  2. $a_{\tau_{k+1}} \in$ *Vicinity*(a_{τ_k}, τ_k) *and* $\tau_i < \tau_k < \tau_{k+1} < \tau_j$*;*
  3. $a_j \in$ *Vicinity*(a_{τ_l}, τ_j)*.*

Figure 1 illustrates direct communication between two agents whereas Figure 2 illustrates indirect communication.

In case of an indirect communication, a_i and a_j may communicate through several agents forming a *daisy chain*. As Vicinity is symmetric and non-transitive, direct communication is symmetric whereas indirect communication is oriented from one 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, $\tau_i = \tau_j$.

3.2 Unfolding the Vicinity relation

In order to compute the next indirect communication between two agents from a given date, Vicinity is projected on a valued-directed-graph $\mathcal{V}$. Formally,

Definition 9 (Vicinity graph) *Let $\mathcal{S}$ be a constellation. The Vicinity graph $\mathcal{V}$ derived form the Vicinity relation is such that $\mathcal{V} = (\mathcal{A}, \{(a_i, a_j)\}, \{\{v_{ij}\}\})$ where:*

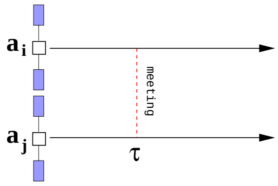

Fig. 1. Direct communication

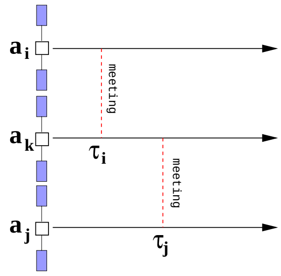

Fig. 2. Indirect communication

– *$\mathcal{A}$ is the set of vertices of $\mathcal{V}$;*
– *edge (a_i, a_j) exists iff $\exists\ \tau \in \mathbb{T}_P$ such that $a_j \in$ Vicinity(a_i, τ);*
– *each edge is labelled with set $v_{ij} = \{\tau \in \mathbb{T}_P : a_j \in$ Vicinity$(a_i, \tau)\}$.*

The following example illustrates this definition.

Example 1 *Let a_1, a_2, a_3 be three agents. Let us suppose that Vicinity is defined as follows on period $P = 20$. The Vicinity graph is shown on Figure 3.*

$$\begin{cases} \text{Vicinity}(a_1, 2) = \{a_2\} \\ \text{Vicinity}(a_2, 5) = \{a_3\} \\ \text{Vicinity}(a_3, 8) = \{a_1\} \\ \text{Vicinity}(a_1, 12) = \{a_2\} \\ \text{Vicinity}(a_2, 15) = \{a_3\} \\ \text{Vicinity}(a_3, 16) = \{a_1\} \end{cases}$$

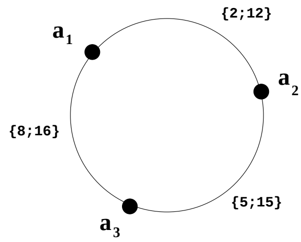

Fig. 3. Vicinity graph for Example 1

Intuitively an indirect communication from agent a_i to agent a_j is a path from vertex a_i to vertex a_j. Thereby from this multi-valued graph, a single-valued graph with respect to the current date is unfolded and the lowest weighted path between both vertices is computed. This single-valued graph is built as it is explored. In order to do that, we propose a modified Dijkstra's algorithm where: (1) the current time τ_i is stored in vertex a_i (initial time plus the weight of the current path); (2) the weight of each edge (a_i, a_j) is computed online as follows: $\min v_{ij} - \tau_i \pmod P$.

Example 2 *Let us resume Example 1 and apply the algorithm in order to compute at time 1 the next indirect communication from a_1 to a_3.*

1. *Consider the edges from the vertex a_1, (a_1, a_2) and (a_1, a_3). The weights of edges (a_1, a_2) and (a_1, a_3) are respectively* $(\min(2 - 1 \pmod{20}), 12 - 1$

(mod 20))), *that is to say* 1, *and* (min(8 − 1 (mod 20), 16 − 1 (mod 20))), *that is to say* 7. *The current times for vertex* a_2 *and* a_3 *are respectively* 2 *and* 8*;*
2. *As a path from* a_1 *to* a_3 *has been computed thanks to the edge* (a_1, a_3)*, a first solution has been found: a direct communication at* $(8, 8)$*.*
3. *Let us continue the exploration from vertex* a_2 *and consider edge* (a_2, a_3)*. Its weight is computed as* (min(5 − 2 (mod 20), 15 − 2 (mod 20)))*, that is to say* 3 *and the current time stored in vertex* a_3 *is* 5*. A new path from* a_1 *to* a_3 *has been computed through the edges* (a_1, a_2) *and* (a_2, a_3)*. A better solution has been found: an indirect communication at* $(2, 5)$*.*

Because Vicinity is common knowledge within the constellation, each agent can compute all communications itself.

3.3 An epidemic protocol

An epidemic protocol based on overhearing [13] has been proposed in [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 $\tau_{K^{\tau}_{a_i}}$.

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 [15]. During a communication round, each agent executes a communication step that has a polynomial complexity in the number of agents and tasks [3].

The agents reason on common knowledge about intentions. Because of the communication delays, this common knowledge concerns only a subset of agents. Formally,

Definition 10 (Common knowledge) *At time* τ*, agent* a_i *knows that agent* a_j *knows intention* $I_t^{a_i}$ *captured by* $K^{\tau}_{a_i}$ *iff:*

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

3.4 The trust model

As indirect communications take time and proposals can be revised meanwhile, some agents' common knowledge may become obsolete. Therefore trust erosion has to be modelled according to the system dynamics. Our application can be viewed as an ad-hoc network, however trust literature on ad-hoc networks [14, 24, 28] focus on the reliability of a node itself and the way to route reliable information. In our application, as agents are trustworthy, trust erosion does not come from the nodes themselves but from interactions between nodes. Consequently we propose a trust model based on communication in order to define a trusted common knowledge.

Last confirmation When two agents communicate at time τ, the agent that receives a given proposal 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 intentions, accordingly. The more time between the generation of a given proposal and the realization date, the less an agent can trust this proposal. However a further confirmation transmitted by the agent that has generated this proposal increases the associated trust again.

As the agents are honest and cooperative, an indirect communication (which is a testimony) is trustworthy itself. Thereby an agent a_i considers that a given proposal generated by an agent a_j has been confirmed if a_j communicates (directly or not) with a_i without modifying its proposal. The last confirmation date is defined as follows:

Definition 11 (Last confirmation date) *Let a_i be an agent, $I_t^{a_j}$ a proposal of an agent a_j about a task t known by a_i. The* last confirmation date *of proposal $I_t^{a_j}$ for a_i at time τ is:*

$$\tau^* = \max_{\substack{\tau_{K^{\tau}_{a_i}} < \tau_j \\ \tau_i < \tau}} \{\tau_j : a_j \textit{ communicates } I_t^{a_j} \textit{ to } a_i \textit{ at } (\tau_j, \tau_i)\} \textit{ and } I_t^{a_j} \textit{ is unchanged}$$

Example 3 *Let us resume Example 1. Let us suppose that, at time 15, a_3 computes the trust associated with a proposal of agent a_1 generated at time 7. a_1 communicated directly with a_3 at time 8 then it communicated indirectly with a_3 at time $(12, 15)$ without modifying its proposal. Thereby the last confirmation date is 12 and a_3 knows that a_1 kept its proposal between times 7 and 12.*

Trust Intuitively, the trust associated with a proposal depends on the time between its last confirmation date and its realization date. As the agents do not have a model of the environment, they cannot predict the arrival of new tasks. However as time goes by, 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 date and the realization date. This number is based on Vicinity therefore each agent can compute its own trust in the others' proposals.

Definition 12 (Meetings) *Let a_i be an agent, $I_t^{a_j}$ a proposal known by a_i and τ the current date. Let τ^* be the last confirmation date of $I_t^{a_j}$ for a_i at time τ. The number of agents $M_{\tau^*}^{a_i}(I_t^{a_j})$ agent a_j will meet between τ^* and $rea(I_t^{a_j})$ is:*

$$M_{\tau^*}^{a_i}(I_t^{a_j}) = \Big| \bigcup_{\tau^* < \tau' < rea(I_t^{a_j})} \textit{Vicinity}(a_j, \tau') \Big|$$

Finally, an agent trusts or does not trust a given proposal:

Definition 13 (Trust) *Let a_i be an agent, $I_t^{a_j}$ a proposal known by a_i and τ the current date. Agent a_i* trusts *agent a_j about $I_t^{a_j}$ iff $M_{\tau^*}^{a_i}(I_t^{a_j}) = 0$.*

Example 4 *Let a_i be an agent that knows proposal $I_t^{a_j}$ at time τ. Let us suppose that $M_{\tau^*}^{a_i}(I_t^{a_j}) = 5$. Agent a_i does not trust a_j about this proposal. Let us suppose that a_j keeps its proposal for long enough to confirm it twice. At each confirmation, a_i can compute $M_{\tau^*}^{a_i}(I_t^{a_j})$ again, e.g. 3 and 1, and can trust a_j more.*

We can notice that the trust criterion of Definition 13 is hard: an agent is not trusted if it meets at least another agent before realizing its proposal ($M_{\tau^*}^{a_i}(I_t^{a_k})$ $= 0$). This pessimistic assumption can be relaxed (e.g. $M_{\tau^*}^{a_i}(I_t^{a_k}) \leq 1$).

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 own outcome. In the literature, the methods dedicated to coalition formation are based on the exploration of the lattice of the possible coalition structures [18, 25]. As the agents often have uncertain and (or) incomplete information on the other agents' costs and preferences, they need to use heuristics [12] or trust [19] to evaluate a coalition value and find the optimal structure.

Generally speaking, these methods have two limits.
On the one hand they are often centralized, assuming that all tasks are known by all agents, and they are performed off-line [6, 9, 16, 21] ; or they use an auctioneer (or other kinds of hierarchy) [1, 22] 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, 22] or in a static vicinity [11, 23]. These constraints are associated with a communication cost [27]. 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 until the complete coalition structure is computed and must be realized quickly. Consequently these methods are very sensitive to the system dynamics.

Be that as it may the coalition formation mechanisms are worthwhile for three reasons: (1) agents gather in order to realize a collective task; (2) the short life cycle of coalitions suits to dynamic environments; (3) agents search for efficient solutions under uncertain and (or) incomplete information. Moreover in our application a compound task requires that some agents should realize the subsets of tasks jointly (see Definition 4). 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 the 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 14 (Coalition) *A* coalition *C is a triplet $\langle A, O, \mathfrak{P} \rangle$:*
- *$A \subseteq \mathcal{A}$ 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;*
- *$\mathfrak{P}$ is the set of tasks that are the* power *of the coalition.*

A coalition C may be :
- complete *iff $O \subseteq \mathfrak{P}$;*
- 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. The **collaboration steps** are such that 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:

I a_i organizes the set of tasks $\mathcal{T}_{a_i}^{\tau}$ as a partition $\{\mathcal{T}_1 \ldots \mathcal{T}_h\}$ according to the compound tasks;

Example 5 *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 $\{\{t_1, t_2\}, \{t_3\}, \{t_4, t_5\}\}$.*

II each $\mathcal{T}_i$ is the goal O_i of a single potential coalition; as subsets $\mathcal{T}_i$ are disjoint[1], the number of potential coalitions generated by agent a_i is given by the number of compound tasks a_i knows;

III from agent a_i's point of view, the potential coalition members for subset $\mathcal{T}_i$ are defined as: $A_i = \{a_k \in \mathcal{A} : \exists\, t \in \mathcal{T}_i \,/\, \exists\, I_t^{a_k} \in \mathcal{K}_{a_i}^{\tau}$ such that $I_t^{a_k} \in \{\Box, \Diamond\}\}$

Example 6 *Let us resume Example 5. 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).*

IV then the power of each potential coalition is defined as: $\mathfrak{P}_i = \{t \in O_i | \exists a_k \in A_i : I_t^{a_k} \in \{\Box, \Diamond\}\}$

Notice that this framework defines the current coalition structure from agent a_i's point of view. A potential coalition may be minimal (thus complete), complete and not minimal or incomplete.

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, the priorities of the goal tasks within the incomplete coalitions are increased. 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 initial priority $prio(t)$ of task t remains the same).

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

a_i is incited to join a coalition if and only if the goal of the coalition is to realize a compound task that is partially planned.

As far as singletons $\{t_j\}$ are concerned, two cases may be considered. (1) If t_j is not planned by a_i, it is because it does not satisfy the optimization criteria (Section 2.3). Therefore a_i does not build any coalition concerning t_j and the priority of t_j remains the same. (2) If t_j is planned, the coalition concerning t_j is complete and its priority remains the same.

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

Example 7 *Let us resume Example 5. 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 the 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 incited to give up a given task in order to realize another one, then *ceteris paribus* is not incited 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 15 (Conflict) *Let* a_i, a_j *be two agents and* C *a coalition* $\langle A, O, \mathfrak{P}\rangle$ *such that* $\{a_i, a_j\} \subseteq A$. *There is a* conflict *between* a_i *and* a_j *iff* $\exists\, t \in \mathfrak{P}$ *such that* $I_t^{a_i} \in \{\Box, \Diamond\}$ *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 that* $\tau_{I_t^{a_i}} < \tau_i$ *and* $\tau_j < \min(rea(I_t^{a_i}), rea(I_t^{a_j}))$ *or* a_j *knows agent* a_i*'s intention about* t. *Else it is a* ***hard conflict***.

Example 8 *Let us resume Example 6. 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* $(\Box)$.

In the remainder, given an agent a_i and a task t, we will denote A^* the set of agents with which it is in conflict about task t, $A^+ \subseteq A^*$ the set of agents in soft conflict and $A^- \subseteq A^*$ the set of agents in hard conflict.

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

1. *(soft conflict)* $\forall\, a_j \in A^+$, a_i *knows* $I_t^{a_j}$. *Conversely either* a_j *knows* $I_t^{a_i}$, *or* $\exists\, \tau_i, \tau_j \in \mathbb{T}$ *such that* a_i *communicated with* a_j *at* (τ_i, τ_j) *with* $\tau_{I_t^{a_i}} < \tau_i$ *and* $\tau_j < \min\,(rea(I_t^{a_i}),\, rea(I_t^{a_j}))$. *In both cases the conflict is symmetric.*
2. *(hard conflict)* $\forall\, a_j \in A^-$, a_j *does not know* $I_t^{a_i}$ *and will not know it before date* $\min\,(rea(I_t^{a_i}), rea(I_t^{a_j}))$. *So* a_j *is not and will not be aware of the 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 swiftness, it is better that the agents realize the tasks as soon as possible and use the fewest resources possible (meaning keeping the pictures in the satellite memory for the shortest time possible, i.e. downloading them as soon as possible). Let us aggregate both criteria in a single expertise criterion. Formally:

Definition 16 (Expertise) *Let a_i be an agent and $A^* \subseteq \mathcal{A}$ be a set of agents in conflict with a_i about a task t. Let us note $rea^* = \min_{a_k \in A^* * \cup \{a_i\}} rea(I_t^{a_k})$ the earliest realization date for task t. The* expert *agent a^* for t is defined using the following distance:*

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

Figure 4 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.

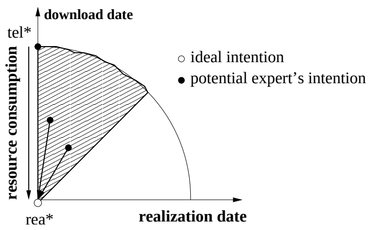

Fig. 4. Expertise criterion

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 the *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 the *collaboration strategy* a_i commits ($\Box$) if it is the expert agent, therefore deciding on a part of the current coalition structure. (3) With the *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 such that $A^- \neq \emptyset$. a_i is aware of the conflict and applies:*

1. *if* $\min\limits_{a_k \in A^-} M^{a_i}_{\mathcal{T}^*}(I^{a_k}_t) > 0$ *then* $I^{a_i}_t \leftarrow \Diamond$
2. *else* $I^{a_i}_t \leftarrow \Box\neg$

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 such that $A^+ \neq \emptyset$. Let rea^* be $\min\limits_{a_k \in A^+} rea(I^{a_k}_t)$. Then agent a_i applies:*

1. *if* $a_i = \arg\min\limits_{a_k \in A_+} ||(rea(I^{a_k}_t) - rea^*, tel(I^{a_k}_t) - rea^*)||$ *then* $I^{a_i}_t \leftarrow \Box$
2. *else let a^* be the expert agent:*
   (a) *if* $M^{a_i}_{\mathcal{T}^*}(I^{a^*}_t) > 0$ *then* $I^{a_i}_t \leftarrow \Diamond$
   (b) *else* $I^{a_i}_t \leftarrow \Box\neg$

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 redundancies. The first metric corresponds to the number of distinct singletons 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 First scenario : dynamic simulations

These experiments are based on a dynamic scenario with 3 agents. Every 6th hour, ground stations send 40 new compound tasks (including at least 2 singleton tasks) to the agents. The number of realized tasks is shown on Figure 5 and the number of realized tasks without redundancies is shown on Figure 6.

Figures 5 and 6 show that informed and coordinated constellations outperform isolated ones. However we can notice that the benefits increase as time goes

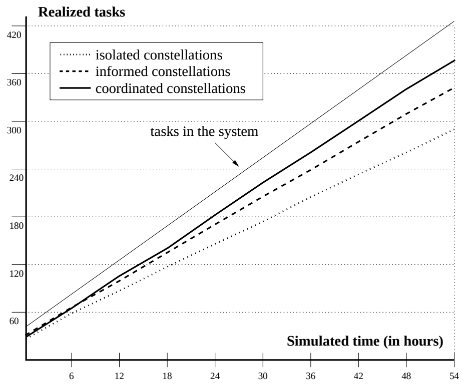

Fig. 5. Tasks

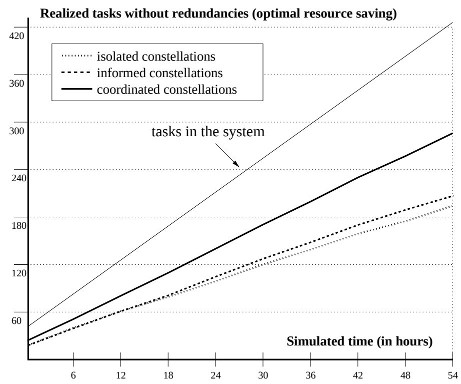

Fig. 6. Tasks with no redundancies

by. Indeed incremental coordination allows coordinated constellations to realize more tasks than the other kinds of constellations. And as time goes by the difference between informed and coordinated constellations increases: incremental coordination allows coordinated constellations to efficiently save and reallocate resources. It is important to notice that, in this experiment, agents are not limited in terms of resources (contrary to real satellites). Consequently the number of realized tasks without redundancies is the main performance measure.

6.2 Second scenario : scalability

In order to experiment the scalability of our system we have considered a scenario with 500 atomic tasks and Walker's satellite constellations [29] of different sizes (1, 4, 6, 8, 9, 12 and 16 satellites dispatched regularly on a finite number of orbital plans). The agents must realize all the tasks and the constellation swiftness and efficiency are then compared.

Definition 17 (Performance) *Let $\mathbb{T}_n$ the time for n agents to realize all the tasks, K the set of realized observations (i.e. the realized tasks and their redundancies) and R the set of realized tasks. The constellation* swiftness *is given by $\frac{\mathbb{T}_1}{\mathbb{T}_n}$ and the constellation* efficiency *is given by $\frac{|R|}{|K|}$.*

We can notice on Figure 7 that the swiftness of isolated constellations is approximated by a logarithmic function whereas the swiftness of informed and coordinated constellations are not regular. This is due to the heterogeneous structure of the satellite interactions. Indeed isolated satellites have no interactions but, for informed and coordinated constellations, interactions exist only between satellites belonging to different orbital plans (see Figure 9).

Consequently 2 satellites situated on 4 plans can have more interactions than 4 satellites situated on 3 plans: the topology of the interactions matters.

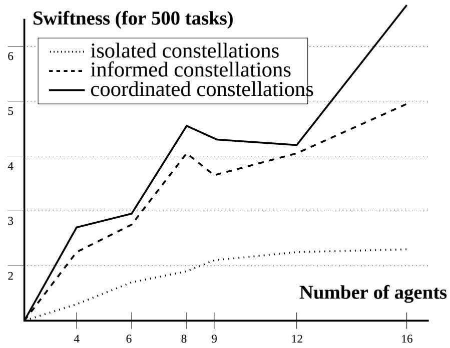

Fig. 7. Swiftness

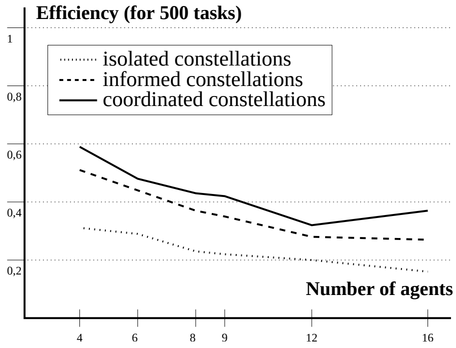

Fig. 8. Efficiency

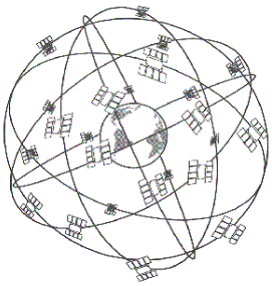

Fig. 9. Different orbital plans

More precisely the number of satellites is not the major parameter but their orbits: few satellites may communicate often whereas many satellites may only communicate from time to time. This phenomenon can be observed between the 8- and 12-satellite constellations.

As far as efficiency is concerned, we can notice on Figure 8 that coordinated constellations are in average 5% more efficient than informed constellations. They are also 19% more efficient than isolated constellations. The constellations are scalable according to Turner [26]: a system is scalable if the resource consumption can be bounded by a polynomial function. In our application, the number of realized observations divided by the number of realized tasks $\frac{|K|}{|R|}$ represents the resource overconsumption: it is the inverse of the efficiency.

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 a trusted common knowledge (in terms of tasks and intentions) as the agents meet. As new tasks appear in the system the agents may revise their intentions. Thereby trust is defined through the communications between agents. Each time an agent communicates, it may receive new information that modifies its intentions. On the other hand the more an agent communicates, the more it can confirm its intentions and the more trust may increase.

The collaboration process is an online incremental decentralized 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. As the agents communicate, they refine the coalition structure dynamically and adapt it to new knowledge.

The experimental results show that the coalition formation mechanism allows the resource consumption to be minimized. Then the saved resources are reallocated in a incremental way and the number of realized tasks is increased. Furthermore our approach is scalable despite the non linear topology of the satellite constellations. It allows to reduce the number of involved satellites (that are highly costly) for the same swiftness. Further works will deal with the possible failures of the agents and their consequences on the collaboration process.

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. Proceedings of the 4th IAT (2004) 162–168
2. A. Brambilla, M. Lavagna, A. Da Costa and A.E. Finzi. Distributed Planning and Scheduling for Space System Flotillas. Proceedings of the 8th ISAIRAS (2005)
3. G. Bonnet and C. Tessier. Collaboration among a satellite swarm. Proceedings of the 6th AAMAS (2007) 275–282
4. 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
5. F. Dignum. Autonomous agents with norms. Artificial Intelligence and Law (1999) Vol. 7:69–79
6. V. Dung Dang, R.K. Dash, A. Rogers and N.R. Jennings. Overlapping coalition formation for efficient data fusion in multi-sensor networks. Proceedings of the 21st AAAI (2006) 635–640
7. S. De Florio, T. Zehetbauer and T. Neff. SCOOP: Satellite Constellations Optimal Operations Planner. Proceedings of the Workshop on Constraint Satisfaction Techniques for Planning and Scheduling Problems at the 16th ICAPS (2006)
8. E. Bornschlegl, C. Guettier, G. Le Lann and J.-C. Poncet. Constraint-based layered planning and distributed control for an autonomous spacecraft formation flying. Proceedings of the 1st ESA Workshop on Space Autonomy (2001)

9. H.J. Goradia and J.M. Vidal. An equal excess negotiation algorithm for coalition formation. Proceedings of the 6th AAMAS (2007) 1052–1054
10. B. Horling and V. Lesser. A survey of multi-agent organizational paradigms. Knowledge Engineering Review (2004) Vol. 19:281–316
11. M. Krainin, B. An and V. Lesser. An Application of Automated Negotiation to Distributed Task Allocation. Proceedings of the 7th IAT (2007) 138–145
12. S. Kraus, O. Shehory and G. Taase. Coalition formation with uncertain heterogeneous information. Proceedings of the 2nd AAMAS (2003) 1–8
13. F. Legras and C. Tessier. LOTTO: group formation by overhearing in large teams. Proceedings of 2nd AAMAS (2003)
14. A.A. Pirzada and C. McDonald. Trust Establishment in Pure Ad-Hoc Networks. Wireless Personal Communications (2006) Vol. 37(1-2):139–168
15. B. Pittel. On spreading a rumor. Journal of Applied Mathematics (1987) Vol. 47:213–223
16. T. Rahwan, S.D. Ramchurn, V. Dung Dang and N.R. Jennings. Near-optimal anytime coalition structure generation. Proceedings of the 21st IJCAI (2007) 2365–2371
17. T. Sandholm. Contract types for satisficing task allocation. Proceedings of the AAAI Spring Symposium on Satisficing Models (1998) 23–25
18. 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
19. S. Sen, I. Goswami and S. Airiau. Expertise and trust-based formation of effective coalitions: an evaluation of the ART testbed. Proceedings of the ALAMAS workshop at the 5th AAMAS (2006)
20. T. Schetter, M. Campbell and D.M. Surka. Multiple agent-based autonomy for satellite constellation. Artificial Intelligence (2003) Vol. 145:147–180
21. 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
22. M. Sims, C. Goldman and V. Lesser. Self-organization through bottom-up coalition formation. Proceedings of the 2nd AAMAS (2003)
23. L.-K. Soh and C. Tsatsoulis. Utility-based multiagent coalition formation with incomplete information and time constraints. Proceedings of the IEEE International Conference on SMC (2003)
24. G. Theodorakopoulos and J.S. Baras. On Trust Models and Trust Evaluation Metrics for Ad-Hoc Networks. IEEE Journal on Selected Areas in Communications (2006) Vol. 24(2):318–328
25. 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
26. P.J. Turner and N.R. Jennings Improving the scalability of multi-agent systems. Proceedings of the Workshop on Infrastructure for Scalable Multi-Agent Systems (2000).
27. 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
28. M. Virendra, M. Jadliwala, M. Chandrasekaran and S. Upadhyaya. Quantifying Trust in Mobile Ad-Hoc Networks. Proceedings of the IEEE KIMAS (2004) 65–71.
29. J.G. Walker. Satellite constellations. Journal of British Interplanetary Society (1984) Vol. 37:559–571.