An Earth Watching Satellite Constellation: How to Manage a Team of Watching Agents with Limited Communications

Sylvain Damiani
ONERA
Centre de Toulouse
2 av. Édouard Belin, BP 4025
31055, Toulouse Cedex 4
France
Sylvain.Damiani@onera.fr

Gérard Verfaillie
ONERA
Centre de Toulouse
2 av. Édouard Belin, BP 4025
31055, Toulouse Cedex 4
France
Gerard.Verfaillie@onera.fr

Marie-Claire Charmeau
CNES
Centre Spatial de Toulouse
18 av. Édouard Belin
31401, Toulouse Cedex 4
France
Marie-Claire.Charmeau@cnes.fr

ABSTRACT

In this paper, we present the problem of management of an Earth watching mission (detection, observation, and tracking of forest fires and volcanic eruptions) by means of a constellation of low-orbit satellites. We show that the mission reactivity requirements and the strictly limited communication means led us to a specific decision architecture. This architecture is based on two components: a tracking task sharing mechanism which is centralized on the ground and regularly activated, and a reactive decision/planning mechanism which is implemented on board each satellite, permanently active, and interruptible at any time. Simulations allow us to compare results obtained with more or less frequent coordinations by the ground. We conclude by showing that this specific application can be seen as an instance of the problem of watching any dynamic unforeseeable system/environment by a team of agents in a setting of limited inter-agent communications.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Dynamic Programming, Plan Execution, Formation, and Generation*

General Terms

ALGORITHMS, EXPERIMENTATION

Keywords

Earth Watching, On-line Planning, Anytime Planning, Coordination

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
AAMAS'05, July 25-29, 2005, Utrecht, Netherlands.
Copyright 2005 ACM 1-59593-094-9/05/0007 ...$5.00.

1. AN EARTH GLOBAL WATCHING MISSION

1.1 Fire and eruption detection, observation, and tracking

The space mission we discuss in this paper has been provided to us by the French Space Agency (CNES) [4]), in order to assess the interest and the feasibility of on-board autonomous planning and scheduling modules. Although it is not an actual mission yet, it is a realistic mission, inspired from the *Bird* (http://spacesensors.dlr.de/-SE/bird/) and *Fuego* [10] projects.

The mission objectives are to *detect*, to *observe*, and to *track* forest *fires* or volcanic *eruptions*. More precisely, starting fires and eruptions must be automatically *detected*, *localized* and roughly *identified*. In case of detection of a fire or an eruption by a satellite, this satellite must immediately send an *alarm* to the concerned ground mission center and trigger an *observation* of the associated ground area. After that and as long as it is necessary, this area must be *tracked* by this satellite and by the other ones of the constellation, i.e. *observed* as regularly as possible. After each observation, data must be *delivered* as early as possible to the concerned ground mission centers.

1.2 A constellation of Earth watching satellites

To fulfill this mission, we assume to have at our disposal the following space and ground physical components:

1. a constellation of 12 identical *low-orbit* (LEO) *satellites*, arranged according to a *Walker* schema [25]: 3 orbital planes, each with an inclination angle of $47,5^\circ$ with regard to the polar axis, 4 satellites per orbital plan, evenly distributed on a circular orbit at an altitude of 700 km;

2. a set of 3 *geostationary* (GEO) *satellites* which together cover the whole Earth surface;

3. a set of ground *mission centers*, possibly dedicated to a specific area and to a specific kind of event (either forest fire, or volcanic eruption).

4. a ground constellation *control center*.

Given their altitude, the LEO satellites have a revolution period round the Earth of about 100 minutes. Figure 1 is a schematic

3D view of the movement of the constellation within a 25 minute period. It represents the trajectory of each LEO satellite within this period. It represents also three ground stations with the cuts of their reception/emission cones at an altitude of 700km: a LEO satellite can receive or emit data from or to a station only when it is inside the associated circle.

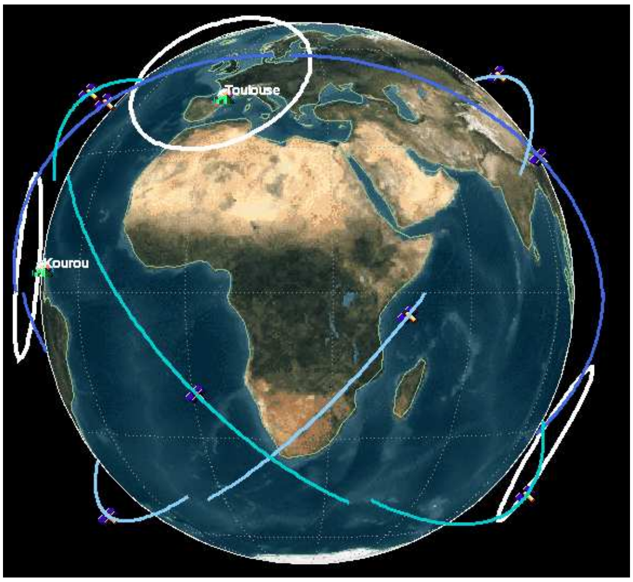

Figure 1: Movement of the constellation within a 25 minute period.

Figure 2 is the 2D ground counterpart of Figure 1. For each LEO satellite, its represents the track on the ground of its trajectory within a 25 minute period and the ground strip that is swept by its detection instrument (see Section 1.3). Note the three orbital planes and the shift between the track of a satellite and the track of the following one in the same orbital plane, due to Earth rotation on itself. Simulations show that the time between two successive flights over a given ground area by any of the satellites of the constellation depends on the area latitude, but is very irregular at each latitude: from some minutes to some hours.

The GEO satellites can be used to relay *alarms* from the LEO satellites to the ground. At any time, each LEO satellite is covered by one of the 3 GEO ones and can thus send an alarm to it. From the GEO satellite, this alarm can be sent to the ground reception station associated with it, and then to the concerned ground mission center via any ground communication network.

The ground mission centers can receive *observation data* from the LEO satellites, but only when they are in visibility of the associated reception station.

The ground constellation control center can send *observation requests* to the LEO satellites, but, as with mission centers, only when they are in visibility of the associated emission station.

1.3 Detection and observation instruments

We assume that each LEO satellite is equipped with two instruments (see Figure 3):

1. an infrared *detection instrument*, whose swath is 2500 km wide. This instrument is permanently active and pointed 30°, that is 400 km, in front of the satellite. Data analysis is instantly performed on board. In case of fire or eruption detection, an alarm is sent to the concerned ground mission center via the currently visible GEO satellite and an observation request is sent to the observation system;
2. an *observation instrument*, whose swath is only 176 km wide. Four observation modes, in the visible, near infrared, and thermal infrared spectrums, are available, according to the kind of phenomenon to observe. This instrument is not permanently active. It is permanently pointed under the satellite, but a mobile mirror in front of it allows it to observe laterally any ground area in the strip that is swept by the detection instrument. Data that result from an observation are not analyzed on-board. They are down-loaded to the concerned ground mission center within visibility windows.

Note that, because the detection instrument is systematically pointed 30° in front of the satellite, there is an one minute delay between the detection of an unexpected phenomenon by a satellite and its possible observation by the same satellite.

Because the satellite can observe a ground area only when it arrives roughly at the same latitude, the starting and ending times of the observation of a given area from a given revolution of a given satellite are fixed and two areas whose latitudes are too close may be *incompatible*: they cannot be observed by the same satellite from the same revolution because of either a temporal overlapping, or an insufficient time to modify the mirror orientation (see Figure 4).

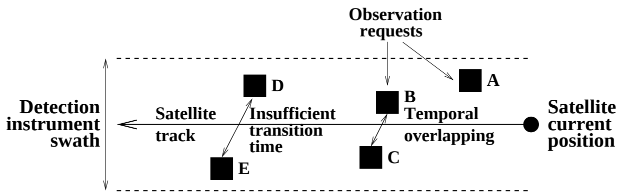

Figure 4: Incompatibilities between observations from the same satellite and the same revolution.

1.4 On-board energy and memory

Each LEO satellite is limited in terms of *energy* and *memory* available on-board. Figure 5 shows the *permanent* and *temporary productions* and *consumptions* of energy and memory that must be taken into account.

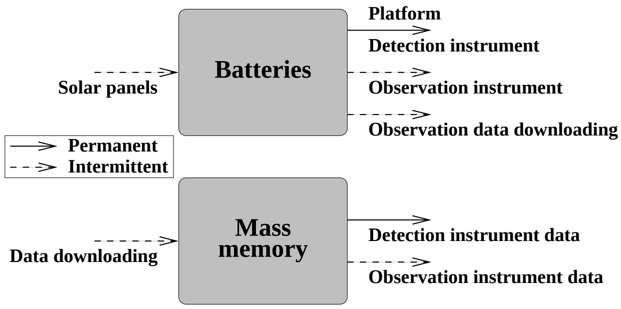

Figure 5: Productions and consumptions of energy and memory.

It is assumed that solar panels are powerful enough to cover the maximum energy consumption during day windows, but enough

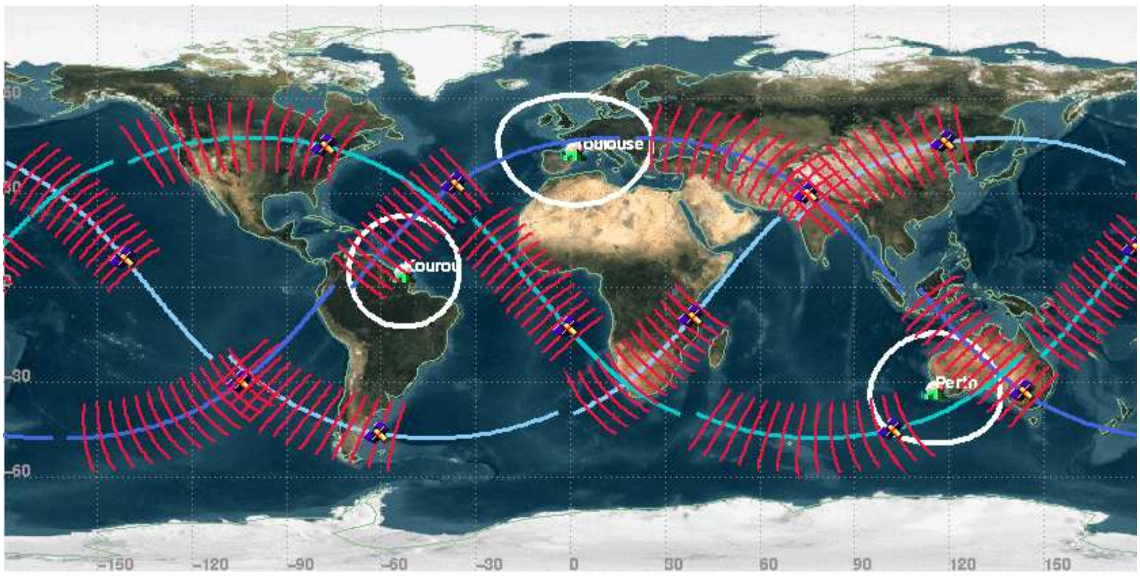

Figure 2: Track on the ground of the 12 satellites of the constellation within a 25 minute period.

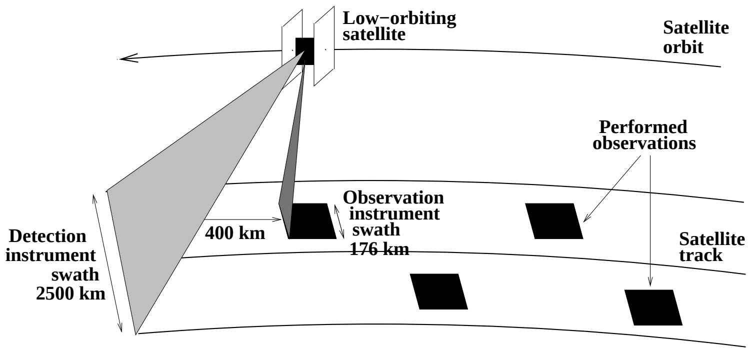

Figure 3: Detection and observation on-board each LEO satellite.

energy must be stocked into batteries to cover night windows. Energy and memory are not independent because observations consume energy and memory and because data down-loading produces memory (by releasing memory space), but consumes energy.

1.5 Observation data down-loading and request up-loading

As previously said, *data down-loading* is possible from a LEO satellite to a ground mission center as soon as the satellite is in visibility of the reception station. But, at any time, a satellite cannot down-load data to more than one station and a station cannot receive data from more than one satellite.

Similarly, *request up-loading* is possible from the ground control center to a LEO satellite as soon as the satellite is in visibility of the emission station. But, at any time, the station cannot send requests to more than one satellite.

1.6 Communication constraints

Figure 6 summarizes the communications that are possible between space and ground components.

It must be stressed that:

- communications between the LEO satellites and the ground, via the GEO satellites, are possible at any time, but limited to unidirectional low rate communications, only able to support alarms in case of fire or eruption detection; such limitations are due to cost and technology constraints: communications between satellites assume that the emitter knows where the receiver is; it is easy if the emitter is the LEO satellite and the receiver the GEO one; it is more difficult in the opposite direction;

- only direct visibility windows between LEO satellites and ground stations can be used for higher rate communications, able to support observation data down-loading and request up-loading.

- no direct communication is possible between LEO satellites;

Let us add that the time between two successive visibility windows between a given LEO satellite and a given ground station depends on the station latitude, but is very irregular along time: from 100 minutes (one revolution) to more than 15 hours.

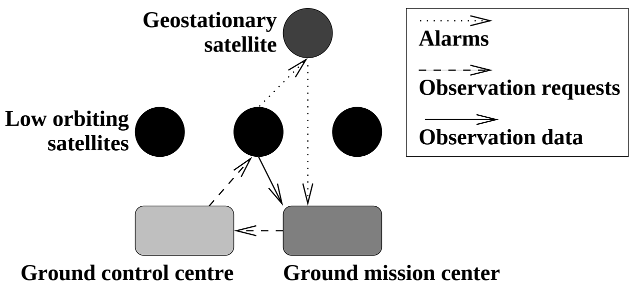

Figure 6: Possible communications between space and ground components.

2. GLOBAL DECISION-MAKING ORGANIZATION

Together, the global *mission objectives* and the *physical/technological setting* that have been presented so far strongly constrain the kind of *decision-making organization* that is possible between the ground constellation control center and the LEO satellites.

The first objective of the mission is to *detect* starting fires and eruptions and, in case of detection, to send alarms, to trigger observations, and to down-load observation data. Between a detection and the observation of the associated area, there is only one minute. Between an observation and the down-loading of the associated data, there is a highly variable time that depends on the next visibility window between the satellite and the concerned ground mission center. This means that the satellite cannot wait for decisions from the ground control center (which could arrive only via visibility windows) to trigger an observation and to down-load associated data after detection. It must be able to make these decisions *autonomously* on-board and to manage for that possible conflicts with previously planned observations and data down-loadings.

The second objective of the mission is to *track* areas where fires or eruptions have been detected, by triggering observations as regularly as possible and by down-loading associated data as early as possible. Because the time between two successive flights of a given satellite over a given ground area can go up to 15 hours, this task cannot be performed by a satellite alone, but by the whole constellation. Thus, tracking must be planned between all the constellation satellites. But, each satellite alone has only a partial view of the current fires and eruptions, and cannot communicate directly with the others. This turns out any choice for a decentralized task sharing mechanism between constellation satellites. In fact, via the alarms that are immediately relayed by the GEO satellites, the ground control center has at any time a complete view of all the fires or eruptions detected by all the constellation satellites. It can consequently *share* tracking tasks among satellites, with however two limitations. The first one is that it may be not aware of the actual state of each satellite, particularly of the actual levels of energy and memory available on-board. The second one is that it has no permanent communication with each satellite and that the result of its sharing will be sent to a given satellite only when this satellite will be in visibility. Both points do not rule out any interest in a sharing of the tracking tasks performed on the ground, but limit its impact: the result of the sharing shall be seen by LEO satellites only as *advice* or *requests*, not as orders; each LEO satellite shall remain able to deal *autonomously* with conflicts between requests coming either from the ground or from on-board detection, by taking into account its actual levels of energy and memory.

The organization that seems to fit the best the physical/technological constraints is thus a mix of *centralized* and *decentralized* decision-making: a *central entity* has at any time a global view of the work to do, shares this work between local entities, and communicates the sharing result to them when it can do it; each *local entity* does at any time the best it can, taking into account its state, the requests/advice of the central entity, and events that may occur unexpectedly.

This setting is strongly different from the one of *Earth observation*, which has been extensively studied for many years, in the setting of individual satellites and in the one of fleets or constellations of satellites [21, 2, 26, 18, 23, 11, 15, 12]. In Earth observation, there is no detection on-board and all the requests come from the ground. This is why there has been no very strong interest in the design of autonomous decision-making capabilities, with only some exceptions [3, 24]: observation plans can be built on the ground and regularly up-loaded to the satellites. But things change as soon as information is produced and analyzed on-board. This is the case when on-board decisions need information about the *actual state* of the satellite which is not accurately available on the ground at the planning time. This is also the case when on-board detection of the *actual cloud cover* allows the satellite to avoid useless observations in the visible spectrum [14], or when rough *on-board image analysis* allows the satellite to remove data associated with unusable observations, and thus to save on-board memory and to avoid useless data down-loading [13, 16]. This is finally the case with *Earth watching*, because of the ability of the satellite to detect *new ground phenomena* and the need for *immediate reaction* in case of detection [5].

3. GROUND SHARING OF TRACKING TASKS

Because requests can be up-loaded to a LEO satellite only when this satellite is in visibility of the ground control center, the control center must prepare requests to this satellite just before a visibility window and for the period between this window and the next one. For this period, it must share tracking tasks among all the satellites, taking into account the fact that, for the other satellites, it will be able to up-load requests to them only later, when they will be in visibility: for these satellites, requests cannot be modified till their next visibility window (see Figure 7).

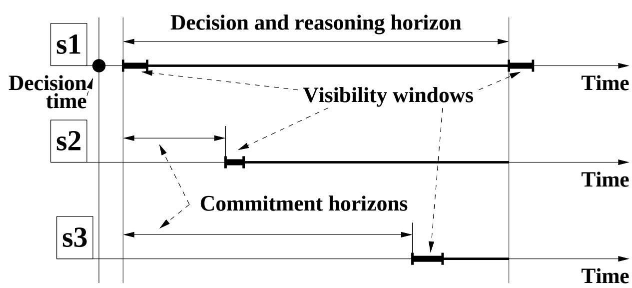

Figure 7: Decision time, commitment, decision and reasoning horizons for the sharing module just before a visibility window for the satellite s_1.

With each *area* a where a fire or an eruption has been detected, is associated a *tracking request* tr. With tr, are associated a *priority level* $p(tr)$, $1 \leq p(tr) \leq p_{max}-1$, a *tracking starting time* $st(tr)$, and a *tracking period* $tp(tr)$. The maximal priority p_{max} is kept for observation requests that result from on-board detection of a new ground phenomenon. Ideally, area a should be observed at each time $st(tr) + i \cdot tp(tr),\ i \geq 0$ and data should be down-loaded immediately after observation. In reality, even by using all the constellation satellites and all the ground stations, one can be only nearing this objective (see Figure 8).

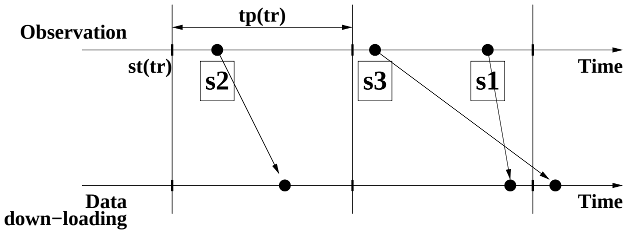

Figure 8: Tracking of a ground area: objective and example of assignment.

The objective is then to assign each *observation o* of each tracking request tr one (or more) pair $< s, r >$, such that satellite s can perform observation o from revolution r, and in such a way that all the tracking requests are satisfied as best as possible : observation times as close as possible to the reference times, data down-loading times as close as possible to the observation times. We associate an *observation note*[1] and a *down-loading note*[2] with each candidate local assignment, i.e. with each triple $< o, s, r >$. Because they are normalized, these notes can be compared and the note of a candidate local assignment $< o, s, r >$ is defined as the *minimum* of its observation note and of its down-loading note. We associate an *evaluation vector*, which is a vector of vectors of notes, with any global assignment: a vector of notes for each priority level from the highest priority to the lowest and a note for each observation of each priority level. We use then an *egalitarist* approach, i.e. *lexicographic* and *leximin* orderings to compare two vectors [3] [9].

Because each candidate local assignment has a specified starting time and duration, an *optimal consistent* global assignment can be computed using a *dynamic programming* algorithm. This algorithm computes recursively the optimal evaluation vector that can be associated with any candidate local assignment i, and represents the best that can be obtained from the beginning of the decision horizon to the ending time of i, by performing i.

[1]Decreasing function of the distance between the observation date and the observation reference date, with the maximum note 1 if the distance is null and the minimum note 0 if it is greater than or equal to $tr/2$.

[2]Decreasing function of the distance between the data down-loading date and the observation date, with the maximum note 1 if the distance is null and the minimum note 0 if it is greater than or equal to a given maximum distance, beyond which we consider the data down-loading utility is null, because information is obsolete.

[3]Let A_1 and A_2 be two candidate global assignments. If we have 2 observations of priority 3 to perform, 3 of priority 2, and 2 of priority 1, evaluation vectors associated with A_1 and A_2 can be for example $\{\{0.3, 0.7\}, \{0.8, 0.2, 0.5\}, \{0.9, 0.7\}\}$ and $\{\{0.3, 0.7\}, \{0.6, 0.6, 0.2\}, \{0.1, 0.5\}\}$. To compare them, we compare both vectors associated with observations of priority 3. They are in this case strictly identical. Thus, we compare both vectors associated with observations of priority 2. To do that, we first order them from the worst note to the best. The result is $\{0.2, 0.5, 0.8\}$ for A_1 and $\{0.2, 0.6, 0.6\}$ for A_2. Then, we compare both ordered vectors lexicographically. They are identical according to the first note, but not according to the second: A_2 is strictly better than A_1 (0.6 *vs* 0.5). This allows us to conclude that A_2 can be preferred to A_1, without considering other notes and other priority levels.

Note that only direct incompatibilities between observations by the same satellite are taken into account by this assignment process. Energy, memory and data down-loading limitation are not taken into account, mainly because the control center does not know at any time the actual levels of energy and memory on-board each satellite.

The result is, for each satellite s, a set $R(s)$ of *observation requests*. With each observation request or, is associated a *priority level* $p(or)$, $1 \leq p(or) \leq p_{max} - 1$, inherited from the associated tracking request tr. In fact, the ground control center sends to a satellite s all the observations it can perform, but all of them that are not in $R(s)$ are systematically assigned the lowest priority 0, which can be interpreted as *to do only if nothing other to do*. Then, if s detects a new ground phenomenon, it is systematically assigned the highest priority level p_{max}. To sum up, whereas the ground control center manages priorities between 1 and $p_{max} - 1$, each satellite manages priorities between 0 and p_{max}.

Note also that the task sharing mechanism should take care of a reasonable *sharing* of the observations to perform between all the constellation satellites, in order to get no overloaded satellite, because an overloaded satellite might be compelled to ignore some ground requests in order to satisfy on-board high priority requests resulting from the detection of new ground phenomena. Inversely, if there are only few tracking requests, the control center could decide to assign each observation more than one satellite, in order to get more frequent observations and above all to be sure that at least one of the satellites performs it successfully.

4. ON-BOARD DECISION-MAKING ORGANIZATION

Because the detection instrument is permanently active and has only one working mode, the only decisions to make on-board are related to the use of the *observation instrument* (to trigger observations), of the *mass memory* (to record or to remove data), and of the *antennas* (to down-load data).

Although the management of observations and the one of resulting data interfere, because observation and data down-loading compete for energy and because data down-loading releases memory for future observation, we present them separately for the sake of clarity. Let us just say that the basic principle we adopted for a common management of observations and data is to give priority to data down-loading for access to energy because it represents a system bottleneck.

4.1 Observation decisions

Let us recall that each LEO satellite is provided at any time with a set of observation requests, coming either from the ground via the visibility windows, or from the on-board detection at any time. With each observation request or, are associated a *priority level* $p(or)$, $0 \leq p(or) \leq p_{max}$, an *energy consumption* $e(or)$, and a *memory consumption* $m(or)$. The starting and ending times of the associated observation is completely determined by the geographical position of the target ground area.

The basic problem is then, just before the starting time of each candidate observation, to decide upon its triggering or not. This decision must be made by taking into account not only the priority of this observation and the ability to trigger it (eventual observation in progress, current mirror orientation, current energy and memory levels), but also the impact of this decision on future possible observations. This implies to reason as far as possible ahead. But, how to set the length of this ahead *reasoning horizon*? Roughly speaking, the larger it is, the more precisely assessed the impact of the current decision is, but the more uncertain data are, and the more time consuming the reasoning process is.

The choice we made is to design an *anytime* reasoning mechanism [27], which adapts itself to the time it has at its disposal. Because candidate observations can be ordered according to their starting time, the reasoning horizon at step i is the sequence made of the first i candidate observations and the problem is to extract from this sequence a optimal consistent sub-sequence. When the reasoning process is started or restarted, it begins with a horizon of length 1: only the next candidate observation is taken into account. When reasoning at step i is finished and time is still available for reasoning, the length of the reasoning horizon is incremented: the $(i+1)$th candidate observation is now taken into account.

The main advantage of such an approach is that a decision is available at any time, in fact as soon as the reasoning at step 1 is finished, i.e. very quickly. This decision is at any time the first candidate observation in the last computed optimal consistent subsequence: at step $i-1$ if reasoning is stopped when reasoning at step i. Although this is not always the case [17], we may expect that the quality of this decision increases with the length of the considered horizon, i.e. with the time available for reasoning.

This iterative mechanism is illustrated by Figure 9: after reasoning at step 7, the optimal sequence of observations is $\{1, 4, 6\}$ and the associated decision is 1, but after reasoning at step 8, the optimal sequence of observations is $\{2, 5, 8\}$ and the associated decision is now 2.

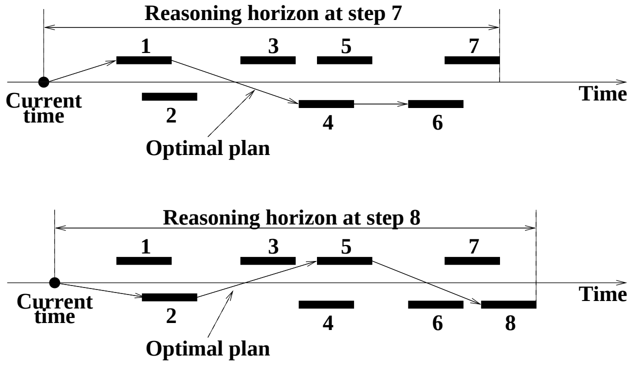

Figure 9: Reasoning on larger and larger horizons.

To compare two consistent sub-sequences, we associate with each sub-sequence an *evaluation vector* that indicates for each priority level p the *number* of selected observations of priority p. Then, we compare both vectors *lexicographically*: an *egalitarist* approach similar to the one used for the ground task sharing.

Provided that energy and memory levels are discretized, the whole iterative mechanism can be implemented using a *dynamic programming* algorithm, close to the one used for the ground task sharing. This algorithm computes recursively the optimal evaluation vector that can be associated with any candidate observation i, any possible level of energy e, and any possible level of memory m, and represents the best that can be obtained from the current time to the ending time of i, by performing i and going out of it with an energy level e and a memory level m.

The most interesting features of this approach is that it is both *anytime* and *incremental*: each time reasoning at step i is finished, a decision is available and executable if necessary; if time is still available for reasoning, reasoning at step $i+1$ can start by reusing everything that has been computed at step i.

4.2 Data down-loading decisions

The problem is, just before the starting time of each visibility window w, to decide upon the data that will be down-loaded to the ground mission center c that will be in visibility. For the sake of clarity, let us consider the case of an isolated visibility window (no overlapping with another visibility window). Let O be the set of observations whose data is currently memorized on-board and is dedicated to c. With each $o \in O$, are associated a *priority level* $p(o)$, $0 \leq p(o) \leq p_{max}$ and a *down-loading duration* $d(o)$. It may be also interesting to consider the *down-loading note* $dn_1(o)$ of o if it would be down-loaded in w and its down-loading note $dn_2(o)$ if it would be down-loaded in the next visibility window of c. The problem is to extract from O a consistent optimal selection $O' \subseteq O$, i.e. a mono-dimensional *knapsack* problem[4].

To compare two consistent selections, we associate with each selection $O' \subseteq O$ an *evaluation vector* similar to the one used for the ground task sharing, i.e. a vector of vectors of notes, which associates a down-loading note with each candidate down-loading: $dn_1(o)$ if $o \in O'$ and $dn_2(o)$ otherwise. Then, we compare both vectors using *lexicographic* and *leximin* orderings: an *egalitarist* approach similar to the one used for the ground task sharing and for the observation selection.

With this egalitarist criterion, the knapsack problem can be solved optimally by using a greedy algorithm, i.e. without any backtrack. This algorithm considers observations in the decreasing order of their priorities and, for a given priority level, in the increasing order of the note $dn_2(o)$. It inserts them one after the other, provided the sum of the durations does not exceed the duration of the visibility window.

5. EXPERIMENTS

Concerning observation decisions, first experimental results have been presented in [6]. But, we carried out since then more ambitious and realistic experiments, involving the whole constellation (12 satellites), a control center, and two mission centers (one dedicated to forest fires and the other one to volcanic eruptions), over a temporal horizon of 16 hours. With regard to the results presented in [7], the results presented in this paper focus on the way of managing the whole constellation.

We assume (1) about 80 ground areas to track that are known at the beginning of the simulation horizon and (2) about 20 that appear during the simulation horizon. These areas are, in both cases, fifty-fifty shared between forest fires and volcanic eruptions and between priorities 1 and 2[5]. For each area of priority 1 (resp. priority 2) , one observation is required every 4 hours (resp. 2 hours). This results in 462 observations requests over the simulation horizon, each one being individually achievable, with about one third of priority 1 and two thirds of priority 2.

We compared three ways of managing the constellation:

1. to share the current tracking tasks among all the constellation satellites at the beginning of the simulation horizon, and send them resulting observation requests; to do it again each time a satellite is in visibility of the control center and new

[4]In the case of overlapped visibility windows, the problem to solve is no more a simple *knapsack* problem, it becomes a *scheduling problem* because not all the data down-loading orderings are consistent with the visibility windows.

[5]For these experiments, we assume that $p_{max} = 3$. Hence, the ground control center manages priorities 1 or 2, but each satellite s manages priorities between 0 and 3: 0 for those that have not been assigned to s by the ground control center and 3 for those that result from on-board detection of a new ground phenomenon.

tracking tasks have appeared, and send it resulting observation requests (SC for *Strong Coordination*);

2. to share the current tracking tasks among all the constellation satellites at the beginning of the simulation horizon and send them resulting observation requests; to do nothing after that (WC for *Weak Coordination*);

3. to do nothing on the ground and to let the constellation satellites detect ground phenomena and generate themselves observation requests (NC for *No Coordination*).

In the first management option, we assume that the sharing of the current tracking tasks among the constellation satellites results in observation requests of priority 0, 1, or 2 and that any new detection by any satellite results on-board in observation requests of priority 3, until a priority 0, 1, or 2 be assigned to the observation of this area by the ground control center and sent to the satellite via a visibility window. In the second management option, things are similar, except that, because visibility windows are not used, observation requests that result from on-board detection remain with priority 3 until the end of the simulation horizon. In the third management option, because there is no sharing, all the observation requests result from on-board detection and remain with priority 3 until the end of the simulation horizon.

For each management option, we measured, for each priority level, the quality of the tracking according to two criteria: (1) the mean note[6] that is obtained over all the observations of all the areas to track, and (2) the number of null notes, i.e. of observations that were required and individually achievable, but have been performed by no constellation satellite (see Table 1).

Priority levels	2		1	
	mean note	number of null notes	mean note	number of null notes
SC	0.507	14	0.515	16
WC	0.498	21	0.513	18
NC	0.450	38	0.508	16

Table 1: Simulation results with three management options.

Concerning the number of null notes for observations of priority 2, table 1 shows a clear increase from SC to NC. Concerning the number of null notes for observations of priority 1, it shows a relative stability. Hence, we have a clear increase concerning the total number of null notes. The same kind of evolution can be observed, concerning the mean notes. These results are not surprising, but allow us to measure the actual positive impact of coordination, particularly for observations of priority 2: tracking tasks are correctly shared between satellites and tracking tasks of priority 1 and of priority 2 are distinguished.

Note that other management options could be profitably considered and assessed: at one extreme, a *Very Strong Coordination* with the assumption of permanent inter-satellite communications and, at the other extreme, a variant of *No Coordination*, where each constellation satellite uses its knowledge about the orbits and the abilities of the other ones to assign a priority to any observation request coming from on-board detection.

[6]Let us recall that the note of an observation of a given area by a given satellite is the minimum of its observation note and of its data down-loading note (see Section 3). If an observation have been performed by several satellites, the note of the observation is the maximum over all the satellites that performed it.

6. DISCUSSION

From the point of view of the Multi-Agent System community, the setting we presented in this paper may seem to be very specific: no communication is directly possible between watching agents; except alarms that can be emitted at any time, only intermittent communications are possible between watching agents and the coordinating agent.

It is undeniable that the decision-making organization we presented in this paper is strongly dependent on this setting. If direct communications would be possible between watching agents via for example geo-stationnary satellites and permanent inter-satellite links (*ISL*), other organizations should be certainly considered [1, 8, 19, 20].

But, there are many real settings where communications are limited for *physical*, *technological*, *financial*, or *security* reasons. Here are some examples of limitations from space or air applications:

- engines on a planet (Earth, Mars . . .) and other ones orbiting around it: intermittent visibility;
- Earth control centers and engines located far in the solar system: intermittent visibility and communication time;
- teams of rovers exploring a planet: intermittent visibility and limited power on-board each engine;
- teams of unmanned aerial vehicles (*UAV*): intermittent visibility and risk of detection in military missions.

In all these settings, there are at one and the same time a need for a *coordination/planning/sharing* of the activities of the various engines and a need for *autonomous decision-making* capabilities onboard each engine (see [22] for a formalization of close settings). On the one hand, the coordination, *a priori* performed on-board one of the engines or in a control center, must take into account both *information* at its disposal and *times* at which orders/advice will be sent to engines. On the other hand, on-board decision-making must be able to make a decision at any moment, taking into account *orders/advice* coming from coordination, as well as the most *up-to-date information* about the engine itself and its environment.

7. ACKNOWLEDGMENTS

Thanks to Michel Lemaître, Félix Ingrand, Solange Lemai, and Frédérick Garcia for stimulating discussions about this work, and more generally about the best way of controlling a system in a dynamic uncertain environment.

8. REFERENCES

[1] A. Barrett. Autonomy Architectures for a Constellation of Spacecraft. In *Proc. of the 5th International Symposium on Artificial Intelligence, Robotics, and Automation for Space (i-SAIRAS-99)*, pages 291–296, Noordwijk, The Netherlands, 1999.

[2] E. Bensana, M. Lemaître, and G. Verfaillie. Earth Observation Satellite Management. *Constraints*, 4(3):293–299, 1999.

[3] E. Bornschlegl, P. David, and H. Schindler. On-Board Software and Operation for PROBA Small Satellite Autonomy. In *Proc. of the International Conference on Data Systems in Aerospace (DASIA-98)*, Athens, Greece, 1998.

[4] M. Charmeau. Mission de Référence pour le DE Autonomie. Note technique DTS/AE/SEA/IL/02-112, CNES, 2002.

[5] S. Chien, R. Sherwood, D. Tran, B. Cichy, G. Rabideau, R. Castano, A. Davies, R. Lee, D. Mandl, S. Frye, B. Trout, J. Hengemihle, J. D'Agostino, S. Shulman, S. Ungar, T. Brakke, D. Boyer, J. VanGaasbeck, R. Greeley, T. Doggett, V. Baker, J. Dohm, and F. Ip. The EO-1 Autonomous Science Agent. In *Proc. of the 3rd Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-04)*, New York City, USA, 2004.

[6] S. Damiani, G. Verfaillie, and M.-C. Charmeau. An Anytime Planning Approach for the Management of an Earth Watching Satellite. In *Proc. of the 4th International Workshop on Planning and Scheduling for Space (IWPSS-04)*, Darmstadt, Germany, 2004.

[7] S. Damiani, G. Verfaillie, and M.-C. Charmeau. A Continuous Anytime Planning Module for an Autonomous Earth Watching Satellite. In *Proc. of the ICAPS-05 Workshop on* Planning under Uncertainty for Autonomous Systems, Monterey, CA, USA, 2005.

[8] S. Das and C. Wu. Enhanced Satellite Constellation Operations via Distributed Planning and Scheduling. In *Proc. of the 6th International Symposium on Artificial Intelligence, Robotics, and Automation for Space (i-SAIRAS-01)*, Montreal, Canada, 2001.

[9] D. Dubois, H. Fargier, and H. Prade. Refinements of the Maximin Approach to Decision-making in a Fuzzy Environment. *Fuzzy Sets and Systems*, 81(1):103–122, 1996.

[10] D. Escorial, I. Tourne, and F. Reina. FUEGO: A Dedicated Constellation of Small Satellites to Detect and Monitor Forest Fires. In *Proc. of the 3rd IAA Symposium on Small Satellites for Earth Observation*, Berlin, Germany, 2001.

[11] J. Frank, A. Jónsson, R. Morris, and D. Smith. Planning and Scheduling for Fleets of Earth Observing Satellites. In *Proc. of the 6th International Symposium on Artificial Intelligence, Robotics, and Automation for Space (i-SAIRAS-01)*, Montreal, Canada, 2001.

[12] A. Globus, J. Crawford, J. Lohn, and R. Morris. Scheduling Earth Observing Fleets Using Evolutionary Algorithms: Problem Description and Approach. In *Proc. of the 3rd NASA International Workshop on Planning and Scheduling for Space*, Houston, TX, USA, 2002.

[13] L. Khatib, J. Frank, D. Smith, R. Morris, and J. Dungan. Interleaved Observation Execution and Rescheduling on Earth Observing Systems. In *Proc. of the ICAPS-03 Workshop on "Plan Execution"*, Trento, Italy, 2003.

[14] J. Lachiver, J. Laherrère, I. Sebbag, N. Bataille, and T. Bret-Dibat. System Feasibility of Onboard Clouds Detection and Observations Scheduling. In *Proc. of the 6th International Symposium on Artificial Intelligence, Robotics, and Automation for Space (i-SAIRAS-01)*, Montreal, Canada, 2001.

[15] M. Lemaître, G. Verfaillie, F. Jouhaud, J.-M. Lachiver, and N. Bataille. Selecting and scheduling observations of agile satellites. *Aerospace Science and Technology*, 6:367–381, 2002.

[16] R. Morris, J. Dungan, J. Frank, L. Khatib, and D. Smith. An Integrated Approach to Earth Science Observation Scheduling. In *Proc. of the 3rd NASA Earth Science Technology Conference (ESTC-03)*, University of Maryland, USA, 2003.

[17] J. Pearl. On the Nature of Pathology in Game Searching. *Artificial Intelligence*, 20:427–453, 1983.

[18] J. Pemberton. Towards Scheduling Over-constrained Remote-sensing Satellites. In *Proc. of the 2nd NASA International Workshop on Planning and Scheduling for Space*, pages 84–89, San Francisco, CA, USA, 2000.

[19] R. Richards, R. Houlette, and J. Mohammed. Distributed Satellite Constellation Planning and Scheduling. In *Proc. of the 14th International Florida Artificial Intelligence Research Society Conference (FLAIRS-01)*, Key West, FL, USA, 2001.

[20] T. Schetter, M. Campbell, and D. Surka. Multiple Agent-based Autonomy for Satellite Constellations. *Artificial Intelligence*, 145(1-2):147–180, 2003.

[21] R. Sherwood, A. Govindjee, D. Yan, G. Rabideau, S. Chien, and A. Fukunaga. Using ASPEN to Automate EO-1 Activity Planning. In *Proc. of the IEEE Aerospace Conference*, 1998.

[22] P. Stone and M. Veloso. Task Decomposition, Dynamic Role Assignment, and Low-Bandwidth Communication for Real-Time Strategic Teamwork. *Artificial Intelligence*, 110(2):241–273, 1999.

[23] M. Vasquez and J. Hao. A Logic-constrained Knapsack Formulation and a Tabu Algorithm for the Daily Photograph Scheduling of an Earth Observation Satellite. *Journal of Computational Optimization and Applications*, 20(2):137–157, 2001.

[24] G. Verfaillie and E. Bornschlegl. Designing and Evaluating an On-line On-board Autonomous Earth Observation Satellite Scheduling System. In *Proc. of the 2nd NASA International Workshop on Planning and Scheduling for Space*, pages 122–127, San Francisco, CA, USA, 2000.

[25] J. Walker. Circular Orbit Patterns Providing Whole Earth Coverage. Tech. Report 70211, Royal Aircraft Establishment, 1970.

[26] W. Wolfe and S. Sorensen. Three Scheduling Algorithms applied to the Earth Observing Systems Domain. *Management Science*, 46(1):148–168, 2000.

[27] S. Zilberstein. Using Anytime Algorithms in Intelligent Systems. *AI Magazine*, 17(3):73–83, 1996.