# Automated Allocation of ESA Ground Station Network Services

**Sylvain Damiani(1), Holger Dreihahn(1), Jörg Noll(1), Marc Niézette(1), and Gian Paolo Calzolari(2)**

(1)VEGA, Aerospace Division
Robert Bosch Straße 7, D-64293 Darmstadt, Germany;
Email: *Firstname.Lastname*@vega.de

(1)European Space Agency, European Space Operations Centre
Robert Bosch Straße 5, D-64293 Darmstadt, Germany;
Email: Gian.Paolo.Calzolari@esa.int

## Abstract

The ESTRACK Planning System (EPS) operationally plans the use of the ESAs ESTRACK ground station network. It uses an incremental planning approach to successively build up the ESTRACK Management Plan using feedback provided by its user missions. The system is configured with a specification of the user missions' needs and the networks constituents' capabilities. The planning process matches both elements to create a plan that is free of conflicts and serves the needs of all missions. To build the plan, a constraint network is constructed from the dynamic input to the system and its configuration. The network is built incrementally exploiting the periodic nature of the communication request timings required by the user missions. The resulting CSP contains temporal binary constraints, linear constraints and disjunctions of binary constraints. Its consistency is checked in two steps: first the DTP (Disjunctive Temporal Problem) part is solved and then the remaining linear constraints are checked using Linear Programming algorithms.
Future developments of EPS will include a more sophisticated resource modeling and the inclusion of more missions and external users. Advanced features like active constraints and global optimization are conceivable.

## The ESA ground station network (ESTRACK)

The European Space Agency (ESA) runs a number of ground stations to support its own missions and the missions of industry contractors. 8 stations owned by ESA plus 3 cooperative stations form the basis of the Esa TRACKing network, ESTRACK. It also includes control and communication facilities.

ESTRACK currently supports 10 operational ESA science missions. It provides services for data downlink and the uplink of commands to satellites in orbit. In addition to the regular ESA missions, ESTRACK supports requests from external users (e.g. NASA).

The mission's requests for satellite-to-ground communication are coordinated for the ESTRACK network as a whole. Until now planning and scheduling of ESTRACK was done manually, supported by a set of tools. The SCUT (**S**pacecraft **C**ommitment **U**tilization **T**ool) is an analysis tool that was used to prove the capability of ESTRACK to support a set of missions. A set of rules defined the missions' communication needs. The rules were applied to build a starting ground station allocation plan. They had to be configured to ensure the production of a conflict free plan. Automated conflict resolution or intelligent search algorithms were not used. The starting plan was loaded into a scheduling tool, where an operator had to edit it. The tool was then used to generate the station schedules.

The planning as it was performed covered the needs of the ESTRACK network. In the future, more missions will have to be cared for and the network will grow by the number of stations. In order to coordinate this growing number of users and providers efficiently, an Automated Planning System is called for. An automated system using an intelligent planning algorithm is able to exploit a flexible assignment of communication services to user missions.

In the following sections of this paper, a detailed problem description is presented. Once the scene is set, a model is demonstrated that supports the solution of the given planning problem. Algorithms that are used for the solution of the planning problem are discussed and first operational results are presented. Finally plans for the future development of the system are discussed.

## Planning the ESTRACK network

An Automated Planning System can cope with the future demand of ground station planning. This planning system is called the **E**STRACK **P**lanning **S**ystem (EPS). The planning system has to dynamically assign the missions requests to services that ground stations offer. Instead of assigning a ground station to a mission, the services, a ground station can offer, are identified and parameterized. Any mission that is using the ESTRACK network may request these services. This adds flexibility and robustness to the planning process.

The system is flexible enough, so that a new mission or ground station can be added by changing the software's configuration database. The software will find a ground station that fulfills the communication requirements of the new mission.

Upon the unavailability of a ground station the system should be robust enough to find an alternative station with the required specification.

The ESTRACK network supports different types of missions that have very different communication needs:

- Earth observation missions with frequent short (several per day) communication periods.
- Missions on highly elliptical orbits (astronomy) with long infrequent communication periods.
- Missions situated in one of the Sun-Earth Lagrange points with daily repeated communication.
- Interplanetary missions with very specific needs and features (long one way light time).

Planning strategies have been developed that can cope with these very diverse communication requirements.

EPS defines for each ground station a set of services it offers to the user missions. Some of the services may be mutually exclusive (they cannot be used at the same time) others may be used in one or several instances together with other services. An example of this would be that one mission uses a stations antenna and online equipment to command a spacecraft while a couple of other missions retrieve recorded data from the same station using its backend equipment. The underlying resource model is part of a future development and is discussed in the last chapter of this paper. The first implementation of EPS assumes an exclusive usage of one ground station per mission in the same time slot. This amounts considering that each station has exactly one complete communication processing chain (e.g. one antenna, one I/F equipment, one baseband equipment…) and that all the services of this chain are exclusive. What's more, those services are booked simultaneously when a mission books the station.

## System Context

The EPS [10] is one of the three elements of the **E**STRACK **M**anagement **S**ystem (EMS) [11]. The EMS is one of the building blocks of ESAs EGOS initiative. The **E**sa **G**round **O**perations **S**oftware – EGOS – includes software systems covering all relevant ground systems of a space mission (see: http://www.egos.esa.int/portal/egos-web/index.html).

The ESTRACK Scheduling System (ESS) uses the planning products of EPS to generate ground station schedules from. The ESTRACK Control System distributes schedules to the ground stations, starts and stops them and monitors their execution. The external Interfaces to the EMS are shown in Figure 1 and explained in Table 1.

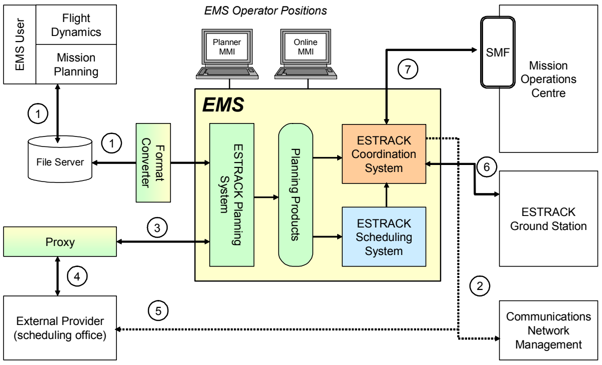

Figure 1 the EMS in its operational context

Table 1 External interface to EMS

| | |
|---|---|
| ① | Submission of mission requests and dynamic mission defined data (event files) Retrieval of views of the **E**STRACK **M**anagement **P**lan (EMP). |
| ② | Transmission of schedules and/or service instance configurations. |
| ③ | Output of relevant portions of the ESTRACK Management Plan; Input of plans received from the external provider. |
| ④ | Interface depending on the external provider, for the **D**eep **S**pace **N**etwork (DSN). Includes submission of the long term request and reception of **S**tation **A**llocation **F**iles (SAF) and **S**even **D**ay **S**chedules (SDS). |
| ⑤ | Transmission of service instance configurations. |
| ⑥ | Schedule distribution, monitoring and control. |
| ⑦ | Operational status information exchange. |

The inputs to the EPS are configuration data and event files, as well as requests and availability plans from external users.

The user missions are configured in a Mission Model that tailors their communication requirements. In the Ground station Model the capabilities of ground stations are stored.

The dynamic inputs to the system are event files which contain mission specific time markers needed to attach activities to. These markers mainly denote the visibility of ground stations but may also express the timings of any other event relevant to a mission (e.g. an operator shift or an illumination condition)

The EPS product is the ESTRACK Management Plan (EMP). Excerpts of the EMP are called plan-views. Plan-views are sent to the user missions as feed-back and should be interpreted by their mission planning systems as station allocation plans. They are also used by the ESS to create the station schedules.

## Planning Process

The planning of the EMS operations is an iterative process between the user missions and the EPS.

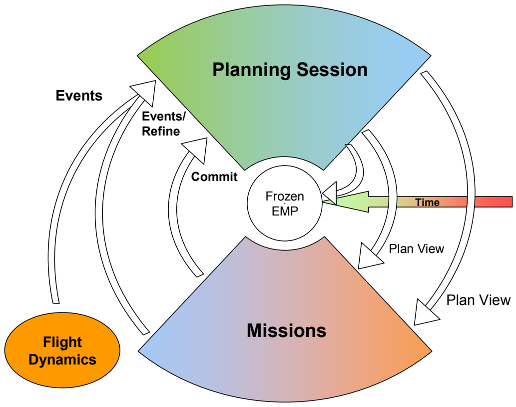

**Figure 2 EPS overall planning cycle**

Figure 2 shows in detail how the planning process starts with the reception and preparation of events from the missions and its flight dynamics systems. From this point on, the figure depicts an inward directed spiral, where the time points inwards and the maturity of the EMP also increases on the inwards direction. On updates to event files the cycle can be restarted at this level. The presence of event files allows for a first planning session and an update of the EMP. A planning session is one run of the rules and algorithms described later in this paper. The result of a planning session is a potentially incomplete set of activities which are added to the EMP.

Plan views are created as defined by the missions and made available to them. The missions can then send refinements. Upon reception of refinements, the EPS may perform a replanning of the affected time range and report the result back to the missions in plan views. There may be several refinement cycles at this level (or none). As a last step, the missions commit the communication segments the EPS has planned for them. Sessions are frozen (i.e. the sessions timings cannot be altered anymore) by the EPS, at a defined time before their scheduling.

Given this planning cycle it is obvious that the planner has to cope with an incomplete set of dynamic input data. As a consequence the EMP cannot be completed for all user missions and is therefore continuously evolving. It contains time ranges where it is completely planned and areas where it is planned only for some of the user missions. Additional planning sessions complete the partly planned areas. Previously planned activities have to be imported into a new planning session, because they constraint the resource usage for this session.

## Planning Objective and Strategy

The aim of the planning process is to produce a valid plan. A valid plan implements the Mission Agreements for all missions on a finite planning period.

The current requirements for the ESTRACK Management System do not call for the creation of an optimized plan. There is no need to find and evaluate several plans as the first valid plan can be chosen.

A number of criteria guide the decision and conflict resolution process. If there is a choice of using an ESTRACK station against an external station (e.g. from the DSN network), the ESTARCK station is used. This is to exploit the ESTRACK network as much as possible.

Some missions require a long continuous contact to ground stations to fulfill their communication requirements. To implement these, it is often necessary to hand over the contact to the spacecraft from one ground station to the next. The planner tries to reduce the number of hand-overs to a minimum.

**Table 2**

| Mission\Station | Santiago | Maspalomas | Kiruna |
|---|---|---|---|
| ERS 2 | 7 | 6 | 5 |
| XMM | 3 | 4 | N/A |
| Cluster | 2 | 1 | 8 |

The case of a concurrent usage of the same ground station service by two missions is called a conflict if the service cannot be shared. To resolve conflicts, the planner uses a priority and preference scheme. The scheme associates to each mission – ground station pair a unique number. An example of this is given in Table 2. The usage of a ground station for a certain mission may also be completely ruled out (denoted by N/A).

If the missions Cluster and XMM tried to use a service of the Maspalomas ground station that cannot be shared at the same time, this results in a conflict. According to the example given in Table 2, the XMM request would have to be moved because it has the lower priority (bigger number).

The scheme used here mixes priorities (as explained) and preferences. It is used for the case where a mission requests a service that can be implemented by two different ground stations. The ground station preferences of one mission are deduced by picking the station with the highest number from the Cluster row. This priority and preference scheme has been used for the first implementation of the EPS.

## EPS –Model

This section provides a logical EPS model and defines the EPS components and their interrelationships. Figure 3 shows the logical EPS model.

Mission specific communication requirements are represented by the Mission Model. The Mission Model contains for all participating missions the information, at which periodic cycles a mission requires a set of ground station services to communicate with a spacecraft. In that context, the ability of a ground station to provide telecommand, telemetry, ranging services etc, is referred to as a ground station service.

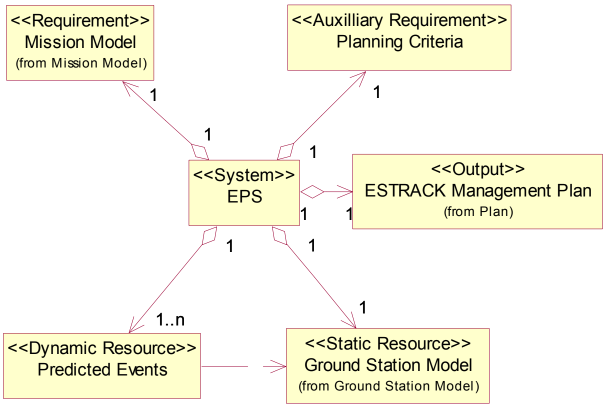

**Figure 3 Overall EPS model**

The ground station model contains all ground stations available for planning. It describes the capabilities of each ground station in terms of services available at each ground station. Ground stations can be seen as static resources, because they do not change during a planning cycle.

The dynamic resource aspect is provided to the EPS in form of predicted events: Among other things, they include the information, at which times a spacecraft is visible from a particular ground station.

Finally, the EPS planning process is guided by Planning Criteria which include but are not limited to ground station preferences and priorities of the missions.

### Mission Model

The EPS planning process is performed to satisfy the requirements expressed in the Mission Model.

The EPS Mission Model can be decomposed as shown in Figure 4. It contains a Mission Agreement for every mission which is planned by the EPS. Each Mission Agreement contains in turn a list of required User Services. A User Service groups the required ground station services and expresses their temporal aspect by a so called Standing Order. The temporal aspect expressed by a Standing Order is the requested periodicity (e.g. every second orbit, twice a week, etc).

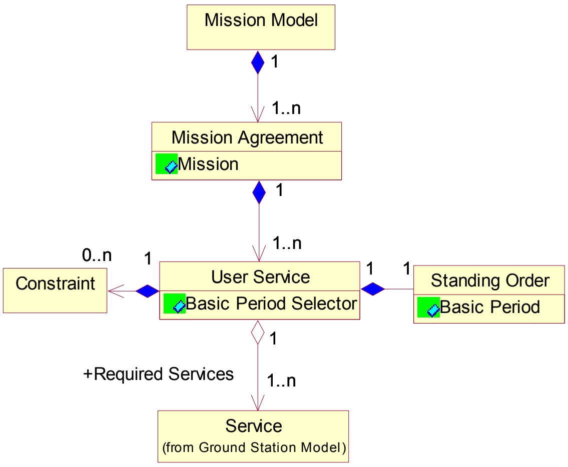

**Figure 4 EPS Mission Model**

In the context of the EPS, the requested periodicity for User Service provisioning is referred to a **B**asic **S**tanding **O**rder **P**eriod (BSOP). BSOPs can be specified based on different units: orbits, hours, days, weeks, and months. The User Service has a Basic Period Selector to select every n$^{th}$ basic period as an implementation interval for the User Service. Figure 5 shows an example for a Basic Standing Order Period of three orbits and a Basic Period Selector of 2. In this example, the User Service would be implemented within the orbits labeled 'Selected BSOPs'.

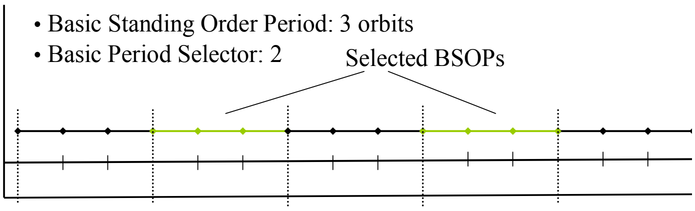

**Figure 5 Basic Standing Order Period**

In addition, a User Service can be constrained. Constraints attached to a User Service affect the way it is implemented: Constraints can have a temporal aspect or can affect the way resources (e.g. ground stations) are used. The following examples of constraints shall provide an idea on how a user service can be constrained:

- The User Service shall be provided for at least 2 and at most for 3 hours.
- A User Service shall be provided 2 times within a Basic Standing Order Period (BSOP). The minimum duration for each service provisioning

is 1 hour, the overall service provisioning within the BSOP shall be four hours.

- Two consecutive provisions of a user service must occur within at least 48 hours but not before 24 hours

## Ground Station Model

Service provisioning is planned based on the ground station resources published by the Ground Station Model. The Ground Station Model as depicted in Figure 6 specifies the available ground stations and their capabilities.

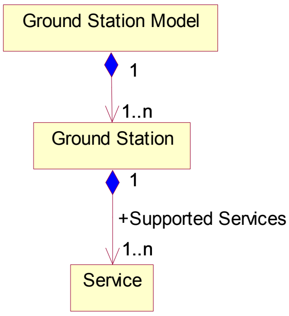

Figure 6 EPS Ground-Station Model

The capabilities of a ground station are expressed as Supported Services. Note that a Service required within a Mission Agreement of the Mission Model can only be instantiated for ground stations which support exactly the required Service.

## ESTRACK Management Plan

Planning results are stored on the ESTRACK Management Plan as so called **O**perational **S**ervice **S**essions (OSS). Operational Service Sessions group Services provided by one ground station for a particular period of time.

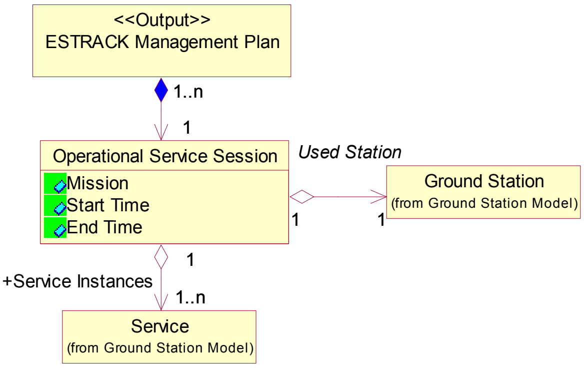

Figure 7 ESTRACK Management Plan

The term Service Instance is used for a Service which is instantiated for a particular time interval. Figure 7 presents the relationship between the various objects stored on the ESTRACK management plan. In the first version of the EPS, all Service Instances have the duration of the Service Session.

## Predicted Events and Service Opportunity Windows

The missions participating in the EPS planning process feed on a regular basis their predicted events into the EPS. In that context, predicted events are:

- **A**cquisition **O**f **S**ignal (AOS) / **L**oss **O**f **S**ignal (LOS) events for the satellite / ground station combinations of a mission
- Start and end of operator shifts
- All other events relevant to planning of ground station allocation

Before the actual planning of ground station allocation is performed by the EPS, a preprocessing of the predicted events is performed. According to mission specific rules, the predicted events are combined to **S**ervice **O**pportunity **W**indows (SOWs). SOWs are periods of time for which the service provisioning for a set of Services is possible. Figure 8 shows an example, how two SOWs are generated: The overlap of ground station / spacecraft visibilities and the operator shift are combined to two SOWs. Each SOW is associated with the ground station providing the service opportunity.

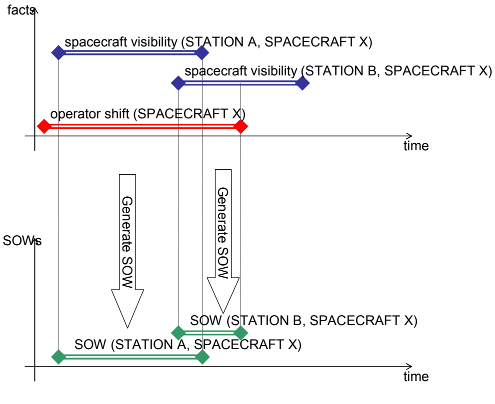

Figure 8 Service Opportunity Window (SOW) generation

The rules on how to create SOWs are expressed as statements formulated in the **L**anguage for **M**ission **P**lanning (LMP). For details on LMP please refer to [12]. SOW generation rules are part of the Mission Agreement of each mission and are associated to User Services. This allows individual SOW generation per User Service. Note that SOW generation rules are not shown in Figure 4 in order not to overload the figure.

## Planning Algorithms

We now present some details on the algorithms used for a planning session. Remember that planning sessions are triggered by the EMS Operators following modifications of the requests or updates concerning spacecraft to ground station visibilities knowledge (see Figure 1). As pointed out before, the aim of a planning session is to assign to each Selected BSOP a set of OSSes (Service Sessions with determined start and end times) implemented on SOWs. These OSSes must be such that all the requirements in the Mission Model and all the resource constraints are respected.

We introduce the concept of Candidate Operational Service Sessions (COSSes) which are OSSes whose start and end times are *time variables*. We say that a BSOP[1] is planned if we have been able to generate a set of COSSes that implement the associated User Service within the corresponding time slot. Once one or more BSOPs are planned, the start and end times of all the associated COSSes constitute the variables of a temporal constraint network. The domains of these variables are determined by the start and end times of the supporting SOWs, while the constraints between those variables are provided by the User Service and the used resources. Given that, a valid plan on the set of planned BSOPs can be generated if and only if the underlying temporal constraint network is consistent.

The mapping between those dedicated concepts and the terms traditionally used in the planning community is the following:

- BSOPs are extended goals generated from the mission model;
- ground stations are unary resources, and SOWs express temporal availabilities of those resources;
- COSSes are performed actions on a partial plan;
- OSSes constitute together an instantiated plan.

**Example 1.** For example, consider the case of two planned BSOPs *B1* and *B2* for two different User Services. *B1* is planned with one COSS *C1* implemented on a SOW *S1*, *B2* with one COSS *C2* implemented on a SOW *S2*. We assume that *S1* and *S2* overlap and are supported by the same ground station. We further assume that the minimum duration of the service for *B1* (resp. *B2*) is $d_1$ (resp. $d_2$). We note $t_A^s$ (resp. $t_A^e$) the start (resp. end) time of the interval *A*. The variables of the underlying temporal constraint network associated to *B1* and *B2* are $t_{C1}^s$, $t_{C1}^e$, $t_{C2}^s$ and $t_{C2}^e$ (the SOWs start and end times are constants). The constraints between those variables are the following:

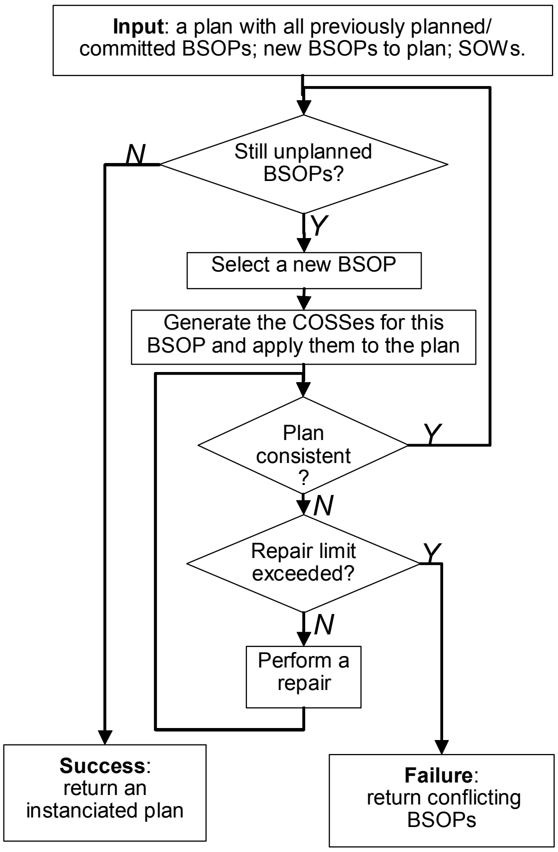

Figure 9 General algorithm

To sum up, the general problem can be decomposed into two basic problems:

- generation of the COSSes for each BSOP
- consistency check of the underlying constraint network

We shall see that the former can be seen as a selection problem and the latter as a scheduling problem. If the constraint network proves to be inconsistent, a *repair* must be performed, by modifying the set of so far generated COSSes.

Figure 10 provides an insight of the global algorithm. It takes as an input new BSOPs to plan and the available

---

[1] For this section we simply call the Selected BSOPs "BSOPs".

SOWs; it extends a plan containing the current implemented COSSes associated to BSOPs planned during a previous planning session. On success, it returns a plan composed of OSSes; on failure, it returns information helping the EPS Operators to take a corrective action. The internal steps are described in the following subsections.

### BSOP selection

Note first that two options were available in order to implement all the BSOPs:

- either plan all the BSOPs, then check the consistency of the global underlying constraint network, and perform some repairs if necessary;
- or plan one new BSOP, check the consistency of the underlying constraint network, performing a repair if necessary, then plan the following BSOP, and so on (*incremental approach*).

We chose the incremental approach essentially in order to make the repairs easier. Thus, at each pass in step “Select a new BSOP” of the general algorithm, a BSOP is heuristically picked and removed from the set of unplanned ones. In our implementation, the BSOPs are ordered by increasing end time and put in a queue (earliest deadline first heuristic), with the hope to limit the extent of the possible repairs to the near past. However this needs to be confirmed by experiments.

### COSS generation

After a BSOP has been selected, we have to generate the COSSes that will be applied to the plan. These COSSes are supported by a subset of the SOWs available for this BSOP and must respect the constraints of the User Service.

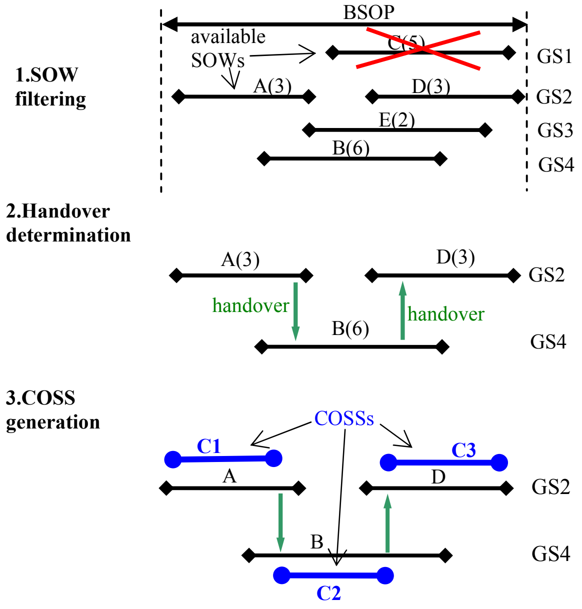

Figure 10 The 3 steps of COSS generation for a BSOP

The first step of the generation is the filtering of all the SOWs that can be proven useless (any COSS generated on them will break some of the constraints specified in the User Service). This is the case when a minimum service distance is required between two consecutive BSOPs of the same User Service: all the SOWs that start too early can be discarded for the COSS generation of the second BSOP.

Secondly, the SOWs that will actually support the COSSes must be selected. Without hand-overs, the selection of the SOW to use for a BSOP would be straightforward: just take the SOW with the highest preference. But the need for hand-overs multiplies the number of possible sets of COSSes. We thus use an optimization algorithm that is able to generate suitable sequences of SOWs that will support the COSSes, taking into account the preferences on the ground stations for the mission as well as most of the constraints and preferences specified in the User Service. This algorithm (not presented here) is based on dynamic programming and has been designed for this specific problem.

From the sequence of SOWs to use for the current BSOP, we obtain the set of COSSes. Then we deduce the new variables and constraints to add to the global temporal constraint network. Note that each constraint is related to a set of time points which are themselves either the start or the end time of a given COSS implemented for a unique BSOP.

Figure 11 shows as an example a BSOP with five available SOWs *A*, *B*, *C*, *D*, *and E*. For each SOW, the corresponding ground station preference level for the mission is indicated between parentheses. During SOW filtering, for some reason *C* is eliminated. Assuming, for example, that the minimum service duration is such that several SOWs are necessary, the handover plan generated at step 2 is the following: first use SOW *A*, then continue on SOW *B*, and finish on SOW *D*. This results in generating three COSSes *C1*, *C2* and *C3* (one for each used SOW), with all associated constraints. We give here the constraints related to the handovers:

$$t^s_{C2} = t^e_{C1} - hd(GS2, GS4)$$
$$t^s_{C3} = t^e_{C2} - hd(GS4, GS2)$$

where *hd(Gsi,GSj)* is the minimum handover duration between ground stations *GSi* and *GSj* for this mission.

### Consistency check

Once the COSSes have been generated for some planned BSOPs, the resulting constraint network must be proven consistent to guarantee that a feasible plan of OSSes can be output.

**Different types of temporal constraints.** The nature of the actual constraints of this network determines the used consistency check method. An analysis of the problem provides three kinds of constraints to be handled:

1. *binary constraints*, of the form $t_i - t_j \leq b$ where $t_i, t_j$ are the variables and $b$ is a constant, widely studied in Simple Temporal Problems [2] (STPs). The consistency check of the associated network is a cubic function of the number of variables. Precedence constraints and minimum handover duration constraints are examples of binary constraints.
2. *linear constraints*, of the form $\sum_i a_i t_i \leq b$, widely studied in Linear Programs [6] (LPs). The consistency check of the associated network is polynomial, and it has the same complexity as the complete solving: it can be seen as solving the phase one of the simplex algorithm[2]. The total service duration is an example of a linear constraint. It defines the sum of the durations all service instances, implemented for one BSOP.
3. *disjunctions of binary constraints*, of the form $\bigvee_k C_k$, where each $C_k$ is a binary constraint, widely studied in Disjunctive Temporal Problems [8] (DTPs) which are NP-complete. These constraints are necessary to express that a unary resource (a ground station for example) can be used by only one COSS at the same time, thus requiring an ordering between the COSSes. The last constraint in Example 1 is an example of a disjunctive constraint.

Note that the general problem is thus to check the consistency of a Disjunctive Linear Program (cons-DLP). However, it is important to stress that in our case binary constraints constitute the majority of the constraints while there are comparatively few linear constraints, and that the disjunctions contain only binary constraints. For a planning horizon of one week, we roughly evaluate the number of time variables to several hundreds, with a few constraints per variable.

**A branch and bound general approach.** One efficient way to solve a disjunctive problem (DTP or cons-DLP) is to check the consistency of a meta Constraint Satisfaction Problem (meta-CSP). The variables of this meta-CSP are the disjunctions, the domain of each variable is the associated set of disjuncts, and the constraints between the variables are implicit [9]. Thus an assignment to some variables is consistent if and only if the associated simple problem (STP or LP) is consistent. The search for a solution consists in the exploration of a tree, each node representing a partial assignment of the meta-CSP. Common CSP solving techniques as well as dedicated ones can be used. Common techniques comprise conflict directed branch and bound and no-good recording. Dedicated ones comprise removal of subsumed variables for DTPs [9] and induced unit clause relaxation for DLPs [3].

Let's consider again Example 1. We further assume that $d_1 = d_2 = 10, t_{S1}^s = 5, t_{S1}^e = 25, t_{S2}^s = 0$ and $t_{S2}^e = 15$.

To check the consistency of the constraint network, we associate a variable *D* of the meta-CSP to the disjunctive constraint, with domain $\{D_1, D_2\}$, where $D_1 : t_{C1}^e - t_{C2}^s \leq 0$ and $D_2 : t_{C2}^e - t_{C1}^s \leq 0$. During the search, *D* is first assigned to $D_1$, but the underlying network is inconsistent, so *D* is then assigned to $D_2$. The underlying network is now consistent, thus a valid plan can be constructed with the planned BSOPs *B1* and *B2* and with associated COSSes *C1* and *C2*. Furthermore, in this plan, *C2* necessarily precedes *C1*.

**Selection of the search strategies.** From an analysis of our problem, we have been able to derive several search strategies to efficiently solve the meta-CSP. Firstly, given that binary constraints are the majority of the constraints, and that STPs are far easier to solve than LPs, a sensible approach is to solve the DTP part of our problem, and check the linear constraints with LP only if a successful leaf is reached. Secondly, in case of failure, we need to pinpoint the set of culprit constraints in order to derive the incriminated COSSes, thus to identify the incriminated BSOPs. Conflict directed strategies are clearly well suited to this as they use discovered conflicts to guide the search. Thirdly, the meta-CSP is a dynamic CSP. Each time a new BSOP is planned and COSSes are generated (resp. a COSS is removed consequently to a repair action), new time variables may be added (resp. removed), thus modifying the implicit constraints of the meta-CSP. New temporal constraints may also be added (resp. removed), thus modifying the pool of variables of the meta-CSP. To cope with this, we will follow advice provided in [7] and experiment no-good recording and oracles.

## Repair

As stated in the COSS generation subsection, COSSes are generated without a guarantee that the former underlying constraint network augmented with the new variables and constraints is consistent. If it not the case, the incriminated COSSes must be detected and a repair action (to modify the COSSes from one BSOP) chosen.

**Identification of the incriminated COSSes.** When the meta-CSP is proven inconsistent the aim, for a repair, is to identify at least one Minimal Unsatisfiable Subset [4] (MUS) of the temporal constraints. A MUS is a set of conflicting constraints such that as soon as one of these constraints is removed, the resulting set is no longer conflicting. In our case, removing a COSS whose start or end time is involved in a MUS enables to solve the conflict identified by this MUS. See [5] and [1] for algorithms to generate MUSes for DTPs and LPs.

**Selection of the COSS to remove.** Among the COSSes identified in a MUS, one must be removed. This choice takes into account general preferences such as mission to ground station priorities in case of a conflict on a resource,

---

[2] Although the simplex algorithm is not polynomial, it is still very efficient.

and heuristics favoring the stability of the network in order to avoid endless repairs.

**Failure report.** The repair process mentioned above is local, thus it is not guaranteed to end with a solution. To prevent an endless repair loop, a stopping criterion is provided, such as a maximum number of repairs, or a maximum time spent in repair. If this limit is reached, the system reports a failure to the EPS Operators together with a set of User Services the degradation of which should allow solving the extracted conflicts.

### Construction of the output

Once all the BSOPs have been successfully planned, it means that a LP, amongst those explored in the meta-CSP tree, has been proven consistent. To obtain a final output plan, it is necessary to fix the start and end times of all the COSSes, thus creating OSSes. To do this, the solution of the LP solved at the end of the last consistency check can be used. Assuming that preferences can be translated in a linear function of the time variables to optimize, it is also possible to solve a last DLP, this time looking for optimality and not only consistency.

## Implementation

The development of the ESTRACK Planning System has been initiated by the European Space Agency in September 2005. EPS is a components of the ESA Ground Operation Software (EGOS), the latest ESA software infrastructure supporting the development of ground systems. The systems will run on PC/LINUX platforms. The core functionality of the systems is developed in C++, and the user interface using ECLIPSE/JAVA/SWT.

In line with the ESA software re-use policy, the implementation of the system is based on reuse of software and design, mostly from the two following sources.

- The Enhanced Kernel Library for Operational Planning Systems (EKLOPS) developed by VEGA for ESA as part of the Mars-Express and Venus-Express mission planning systems (see [12]). This set of libraries supports all aspects of the development of an operational planning system for space mission and provides the core of the planning functionality required for the development of EPS.
- The ESOC Ground Operations Software (EGOS), which provides infrastructure components for generic functionality such as system processes location, system processes management, system processes monitoring and control, system static/dynamic configuration management, services Management Framework (SMF), Users and Privileges management, Events/Alarms management, Files management, and Generic File Transfer.

This approach ensures a safe development of the systems at low cost.

EPS will be ready for operation before the end of 2006.

## Operational Results

The operational results provided below have been obtained with the first operational release of the system. The differences with the features described in the algorithm section are the following:

- the linear constraints are not checked, the consistency check solves a DTP, using an adapted version of Epilitis algorithm [9];
- the choice of the BSOP to modify for conflict resolution is directly based on the no-goods of the meta-CSP returned by Epilitis following the failure, not on MUSes;
- a basic form of oracle is used between two calls of Epilitis: the valid assignment of the meta-CSP obtained after planning a BSOP is the start point of the new consistency check when planning the next BSOP[3]; all no-goods are conserved.

We present results for the planning of the communications for 5 satellites and for 2 weeks. The BSOPs are based on the orbits of those spacecraft (roughly 2 days). During each orbit, for the first satellite (XMM), one contact is required between two absolute dates, and 3 handovers are allowed. For the others (Cluster), 2 contacts are required and no handover is allowed; additionally each contact must last at least 5 hours and must be separated by at least 12 hours from the preceding one.

For this instance, the planning lasted 94 seconds. 219 activities (COSSes), 1533 binary constraints and 25 disjunctive constraints were generated. Only 2 repairs were necessary: this is a small number but fairly representative of current EPS problems which are underconstrained.

As the SSOW generation time was negligible compared to the consistency check, we now focus on the latter. Figure 12 shows the evolution of the consistency check computation time as more and more BSOPs are planned. 4 kinds of behavior of the consistency check can be pointed out: (a) represents successful consistency checks without backtrack, (b) successful with backtrack ones, (c) failed ones, (d) successful ones started from scratch (unlike (a), (b) and (c) which take advantage of the oracle).

The regular rise of the computation time is directly due to the increase of the size of the generated Simple Temporal Networks. We also note that in such loosely constrained problems, proving inconsistency (case (c)) is more difficult

---

[3] Provided that the forward check is successful with those new meta-constraints (see [9] for more details on Epilitis implementation).

than proving consistency (cases (a) and (b)). To finish with, what strikes when comparing cases (d) to the other cases is the benefit of the use of the oracle, whose impact is expected to increase as more and more BSOPs are planned.

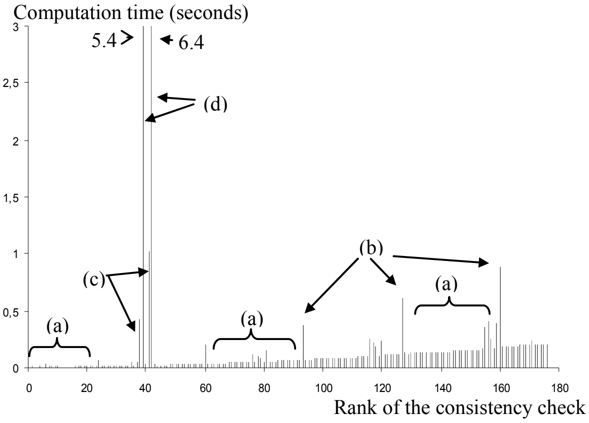

**Figure 11 Consistency check computation time**

## Future Work

The ESTRACK Planning System presented in this paper is the first version of a major component of the ESA ground systems infrastructure, EGOS. The full scope of requirements that was defined for this system is expected to be implemented in later versions of the system. The coming requirements can be parted in three groups. The first group adds the data input mechanisms for a second set of ESA missions. The second group contains the requirements to handle requests from and offers to external users.

The third and biggest group deals with the granularity of ground station services. As mentioned in the introduction to this paper, in the current implementation a ground station can only be used by one spacecraft at a time. The next version of EPS will define discrete resources for ground station that will allow for the parallel use of a ground station by several missions. It will be possible to use different services and/or several instances of the same service at the same time.

In addition to the requirements that are already defined, advanced features may be considered for future versions of the EPS. One is the implementation of active constraints. This concept would support the visual edition of finalized scheduled plans. The edition would be performed by the EPS operator who wants to tailor a specific plan to his needs. The consistent constraint network, which is an intermediate result of the planning process, would be kept with the plan. Each edition the operator performs would cause a reassessment of the constraint network including a possible repair action. Thus the plan would be kept consistent and valid during manual edition.

## References

[1] Chinneck, J., and Dravnieks, E. 1991. Locating Minimal Infeasible Constraint Sets in Linear Programs. ORSA Journal on Computing 3(2):157–168.

[2] Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal Constraint Networks. Artificial Intelligence 49(1-3): 61-95.

[3] Li, H., and Williams, B. 2005. Generalized Conflict Learning For Hybrid Discrete Linear Optimization. Proceedings of the Eleventh International Conference on Principles and Practice of Constraint Programming (CP).

[4] Liffiton, M., and Sakallah, K. 2004. On Finding All Minimally Unsatisfiable Subformulas. Proc. 8th International Conference on Theory and Applications of Satisfiability Testing (SAT-2005) 173–186.

[5] Liffiton, M.; Moffitt, M.; Pollack, M.; and Sakallah, K. 2005. Identifying Conflicts in Overconstrained Temporal Problems. Proceedings of the 19th International Joint Conference on Artificial Intelligence.

[6] Schrijver, A. 1998. Theory of Linear and Integer Programming. John Wiley and Sons.

[7] Schwartz, P., and Pollack, M. 2005. Two Approaches to Semi-Dynamic Disjunctive Temporal Problems. ICAPS Workshop on Constraint Programming for Planning and Scheduling.

[8] Stergiou, K., and Koubarakis, M. 1998. Backtracking algorithms for disjunctions of temporal constraints. Proc. of AAAI-98.

[9] Tsamardinos, I., and Pollack, M. 2003. Efficient Solution Techniques for Disjunctive Temporal Reasoning Problems. Artificial Intelligence 151(1-2):43-90.

[10] EMS Software Requirements Specification, DOPS-ESOC-EMS-SRS-0001-OPS-GIB Issue 1.3, 2005-10-11

[11] Study on ESTRACK Management and Scheduling, Final Report, RN GSS-EMS-SDY-FR-0001, Issue 1.0, 21st February 2005

[12] Noll, J. and Steel, R. 2005. EKLOPS: An Adaptive Approach to a Mission Planning System. Proc. Of IEEE Aerospace conf.