A Planning and Scheduling System to Allocate 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
Europaplatz 5, D-64293 Darmstadt, Germany;
Email: Firstname.Lastname@vega.de

[(2)]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) is a fully integrated planning system dedicated to the automated allocation of ground station services to space missions. Developed for ESA, currently in acceptance phase, it is designated to be operationally run at the European Space Operations Centre in Darmstadt, Germany. It consists in the incremental updating of a global contact activity plan, automatically frozen one week before execution. This global plan is build by the merging of working plans defined for arbitrary temporal ranges. Each plan contains temporal facts representing goals, activities or exogenous events, and constraints linking their start and end times. Goals are created by the automated extension of abstract periodic contact requirements for each mission, while a generic plan query language facilitates the generation of the contact availabilities. During the planning process, each goal is successively planned, i.e. the necessary contact activities are generated and the consistency of the updated global temporal constraint network is checked. In case of conflicts, EPS supports automated repair, degradation of service requirements, and finally provides useful conflict information to the user should the repairs fail.

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. 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 has been developed.

This system, named ESTRACK Planning System (EPS), performs an automated centralized allocation of the ground station services to the space missions. The inputs are contact requirements coming from the missions, and event files including event predictions relevant to the planning, such as visibility windows coming from Flight Dynamics. The control is performed by an operator which creates and commits contact allocation plans. The output (Planning Products) consists of a set of booking periods of the ground stations by the missions with the associated required services. This booking evolves as the event timings are updated and as more and more missions are taken into account. The reader will find more details about the EPS context and the involved planning cycles in [1].

Objectives of the system

The goal of the planning process is to produce a valid plan. A valid plan implements all the mission requirements on a finite planning period.

Table 1 Missions/Stations priorities/preferences

Mission\Station	Santiago	Maspalomas	Kiruna
ERS 2	7	6	5
XMM	3	4	0
Cluster	2	1	8

Note that no global optimization is required. However, a number of criteria guide the decision. In particular, the planner uses a priority and preference scheme which associates to each mission – ground station pair a unique number. An example of this is given in Table 1. This scheme is used to drive the station allocation when two missions compete for the usage of the same ground station at the same time. It is also used when a service can be implemented by two different ground stations. In such a case, the one with the highest preference shall be chosen by the mission. Additionally, a "0" denotes that the ground station cannot be used at all by the mission.

Planning framework

The planning framework is based on the Enhanced Kernel Library for Operational Planning Systems (EKLOPS) [6] developed by VEGA for ESA as part of the Mars-Express and Venus-Express mission planning systems. This set of C++ libraries supports all aspects of the development of an operational planning system for space missions. In particular, it provides the core of the planning functionality required for the development of EPS.
The framework allows building incrementally a so called "master plan" which centralizes all the activities that have been decided for the system and all information taken into account for this purpose. When a new transient plan is created, it is initialized with all information associated to its time range retrieved from the master. When modifications have been applied, the transient plan can be committed to the master plan, which means that the contents of the master are overwritten for its time range. A merging is also performed to avoid inconsistencies at the boundaries.
Like most of the space planning systems, a state based representation of the evolution of the system is used. Thus, plans contain facts (events and activities) with start time, end time and state information. Resource profiles can also be represented, as well as constraints linking the different objects. In EPS, for example, temporal constraints between the facts are incorporated into the plans.
The solving of planning problems is based on the application of rules on the plan. Rules are hard coded, however they can be parameterized through XML configuration files. EKLOPS provides a mechanism to manage complex dependencies between the rules, but only in an acyclic way. Inside the rules, complex processing is performed by translating the plan objects into problems that are tackled by solvers, some developed within the system, others being external free or commercial software. Finally, some rules can be flexibly defined via queries based on the **L**anguage for **M**ission **P**lanning (LMP) [6]. The system interprets and apply LMP queries which define simple operations on the facts of the plan: matching with logical conditions, creation of activities, setting of parameters and timings of the created activities... The main advantage of LMP over the basic rule mechanism is that the code does not have to be modified when an LMP query is modified.

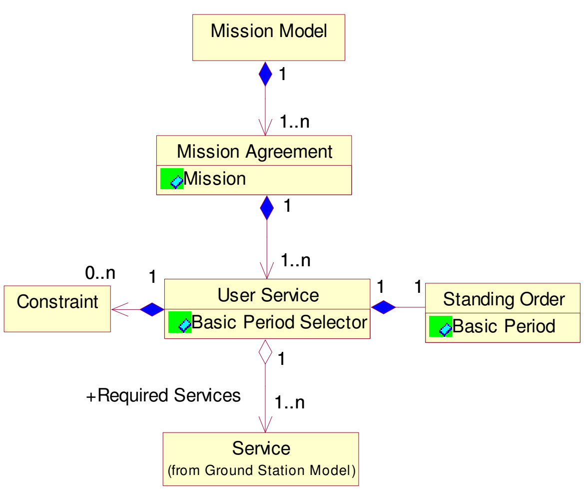

Figure 1 EPS Mission Model

In the sequel, we present the way this framework is used to generate and to solve planning problems.

Generation of planning problems

Modeling of the goals

Figure 1 presents the way the mission requirements are encoded in the configuration database. To satisfy all the mission requirements means that for every mission (Mission Agreement), for every contact requirement (User Service), inside each regularly defined time period (Standing Order), a set of services must be provided taking into account a set of constraints. The constraints define special patterns that must be respected by the activities inside each period and the timing constraints between the activities. Each one of the time range can be considered as a goal, and is referred as BSOP (Basic Standing Order Period) in EPS terminology.

Modeling of the resources

Service provisioning is planned based on the ground station resources published by the Ground Station Model. The Ground Station Model as depicted in Figure 2 specifies the available ground stations and their capabilities.

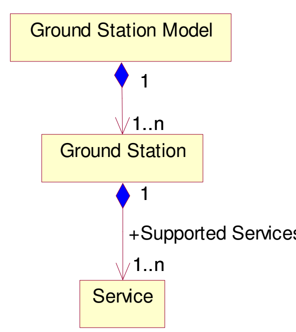

Figure 2 EPS Ground-Station Model

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.
Each ground station can be used by only one mission at the same time: it is a unary reusable resource in the scheduling terminology.

Pre-processing

A planning run is parameterized by a set of missions and the plan range. Before the planning process itself occurs, the goals are created and all the contact availabilities incorporated to the plan.
After goal extension, the plan contains a set of timelines, one per User Service. Each timeline consists of a sequence of BSOPs, which meet (in the interval algebra terminology).
All the contact availabilities, which will support the contact activities, are generated for each User Service. This process utilizes the events (visibilities, operator shifts, etc.) that are already in the plan. LMP (see above) allows taking into account basic timing constraints in order to rule out some of the availabilities, hence to limit the planning search space. In EPS terminology, those availabilities are denoted SOWs (Service Opportunity Windows). Figure 3 gives an example of a SOW generation rule.

```
fact(?id1, ?gs, Vis_el_5, ?start, ?end)
^ parameter(?id1, satelliteId, XMM)
^ fact(?idav, ?gs, GsAvailable, ?astart, ?aend)
^ overlaps(?dur, ?start, ?end, ?astart, ?aend, ?ostart, ?oend)
-> activity( ?newId, sowGenerator, SOW, ?ostart, ?oend)
-> parameter( ?newId, groundStation, ?gs )
```

Figure 3 A SOW generation rule

Planning problems solving

As shown on Figure 4, a planning run consists in iteratively planning all the unplanned BSOPs, which means generating the contact activities such that they match the constraints defined in the User Services. The main steps are briefly described in the sequel of this section.

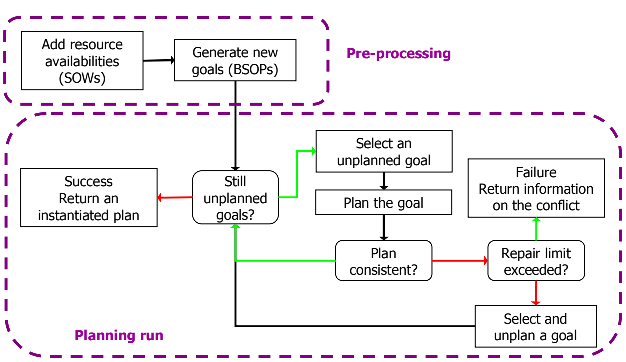

Figure 4 Global planning algorithm

Selection of unplanned goals

The principle of the general algorithm is to incrementally update a consistent plan by iteratively selecting goals from the unplanned goals queue. In the current implementation, this selection uses the earliest deadline first heuristic.

Generation of candidate contact activities

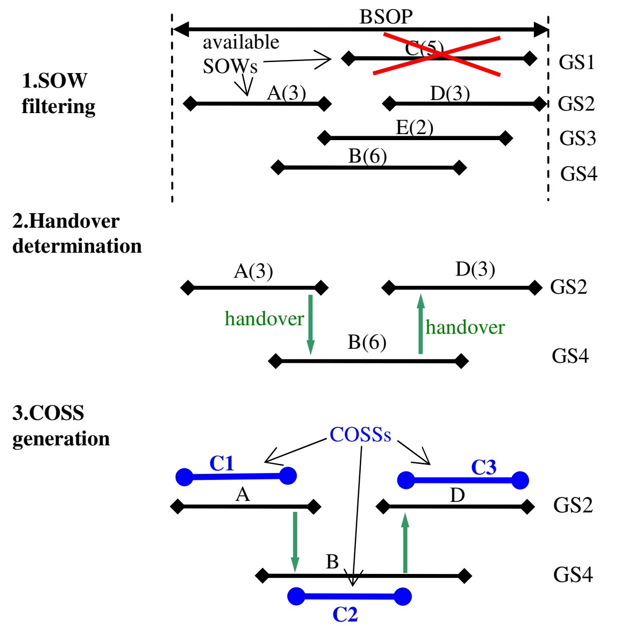

Figure 5 Contact activity generation

Once a goal has been selected, candidate contact activities (COSS for Candidate Operational Service Sessions in EPS terminology) and temporal constraints associated to their start and end times are generated. As shown on Figure 5, this process consists of three steps:

1. enumeration of all the available contact availabilities in the goal time range, and if applicable pre-filtering of the ones that must not be used (e.g. taboo list after a conflict);
2. selection of the availabilities that will support activities and definition of the order in the sequence;
3. generation and addition to the plan of the contact activities and of the temporal constraints; the temporal constraints may link the start and end times of activities and events present in the plan.

The second step uses a dedicated optimization algorithm based on dynamic programming. It basically generates the longest possible path through the available SOWs for the BSOP time range, trying to take into account as many of the constraints defined in the associated User Service as possible. A path is a sequence of time intervals each one associated to a SOW, and may contain discontinuities. The value of such a path is its time length weighted by the preferences of the supporting SOWs.

Consistency check

As described in more detail in [1], the temporal constraint network that is managed by EPS generally defines a Disjunctive Linear Problem [2] or a Mixed Integer Problem. Indeed it contains linear constraints, binary constraints, and disjunctions of binary constraints. However the fact that there are no disjunctions of linear constraint allows solving it by implementing an extension of Epilitis algorithm [5], which initially deals with Disjunctive Temporal Problems (DTP).

The principle of Epilitis is to solve a meta Constraint Satisfaction Problem (CSP) where a variable is associated to each disjunct of the DTP, the domain of a variable is the set of associated disjuncts, and the constraints between assignments of variables are implicit: a partial assignment is valid iff the associated Simple Temporal Problem (STP) [2] is consistent. The search for a solution consists in a conflict directed tree search in the partial assignments of the meta CSP. It uses state of the art CSP solving techniques which fit well with the underlying STP check.

Our extension consists of integrating the checking of the linear constraints for each Epilitis solution. In particular we have implemented and integrated a conflict extraction mechanism based on the phase one of the Simplex algorithm.

Note that the meta CSP is dynamic: each time a new goal is planned (resp. unplanned), temporal constraints are added (resp. removed) from the global temporal constraint network. To cope with this, we have followed advice provided in [4] and implemented no-good recording and oracles. In particular, in our rather under-constrained problem setting, oracles have proved to dramatically increase the speed of the consistency check.

Conflict management

In case of conflict, Epilitis returns the involved meta variables. The meta variables are associated to constraints linking start and end times of some contact activities. One of those activities is chosen for being deleted, either by looking at the mission-ground station priorities, or randomly. As our general planning algorithm is based on a per goal basis, deleting an activity implies to un-plan the whole associated goal, and then to append it to the unplanned goals queue.

A limited number of consecutive repair iterations is allowed. If the limit is reached, then a degradation of one of the involved User Services is locally applied, practically resulting in dropping one or more goals on a given period. In the case when all the goals involved in the conflict are inside an already degraded period, then the planning process stops and a failure report indicates the conflicting activities, and the conflicting goals.

Exporting the solution

Once all the goals have been successfully planned, the start and end times of all the activities are set to their definitive value. This is done by applying the phase two of the Simplex algorithm to the solution node of the meta CSP, using the objective function defined for the generation of the contact activities. Then the operator can get an overview of the plan in the control GUI. If the plan is accepted, the operator can commit it to the master plan.

System usage

The system is still in operational testing. As far as the efficiency is concerned, it for example possible to plan the contacts of 7 spacecraft for 10 days in 800 seconds with 23 repairs, which amounts to 325 activities and 2800 temporal constraints (18 disjunctive constraints). The most time consuming task is the consistency checking of the underlying STP when it is started from scratch in case of repairs.

The log messages and the reports in case of failure help the operator to identify the reasons of the conflicts. He can then try to guide the system for instance by modifying the preference scheme in the configuration database.

To conclude, we have presented a complete and soon fully operational planning and scheduling system. Although designed to cope with a specific problem, it relies on state of the art algorithms which have been extended for the purpose.

References

[1] Damiani, S.; Dreihahn, H.; Noll, J.; Niézette, M., and Calzolari, J.P., 2006. Automated Allocation of ESA Ground Station Network Services. International Workshop on Planning and Scheduling for Space.

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

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

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