Automated Plan Generation for Alphasat Payload Operations

Nicola Policella and **Henrique Oliveira**
ESOC, European Space Agency
Darmstadt, Germany
`firstname.lastname@esa.int`

Edoardo Benzi
ESTEC, European Space Agency
Noordwijk, The Netherlands
`firstname.lastname@esa.int`

Abstract

In 2013 the Alphasat spacecraft will be launched: in addition to its main commercial payload, four Technology Demonstration Payloads (TDPs) will fly on-board. The different payloads are provided by different research institutes, which will be able to define in-orbit demonstration tests of these new technologies. To optimize this opportunity, coordination of the different, possibly conflicting, payload operations is required.

A software system to support the management of the TDP operations has been developed. While this system is intended to be completely automated (i.e., without any human intervention in the nominal case), it has been designed to keep all the different users (system and TDP operators) in the loop. This paper presents this system and in particular focuses on its core: the planning engine.

Introduction

In recent years the European Space Agency has been focusing more and more of its attention on the use of Automated Planning and Scheduling solutions to support space operations. A series of activities have provided different planning systems to support daily operations of different space missions. One of the first examples, if not the first, was the MEXAR2 system (Cesta et al. 2007), a complete planning and scheduling software solution for the MARS-EXPRESS memory downlink problem. This case was then followed by systems developed to support different aspects of the space realm, such as Long-Term planning (Cesta et al. 2011), Science Observation selection (Pralet and Verfaillie 2009; Kitching and Policella 2011), critical mission phases planning (Policella, Oliveira, and Siili 2009), etc. In particular, in order to facilitate the design and implementation of advanced P&S software, the APSI framework was developed (Cesta and Fratini 2008).

The overall impact of planning and scheduling techniques is even bigger when considering the results obtained outside ESA. Just to limit our attention to space related applications we can mention several examples from spacecraft autonomy (Muscettola et al. 1998) to planning & execution (Knight et al. 2001), Earth Observations allocation (Bensana, Lemaitre, and Verfaillie 1999), and so on.[1]

This paper presents an application of planning and scheduling techniques to support the upcoming Alphasat operations.[2] The spacecraft is equipped with four Technology Demonstration Payloads. Since these payloads will be operated by different research institutes, coordination of their activities is required, which is provided by ESA via the TDP ESA Coordination Office (TECO). In particular, the planning system presented in this paper has been designed for managing and coordinating the different payload requests. The final system has been developed using the APSI planning framework, exploiting some of its general modeling and solving functionality of the framework, and integrating ad-hoc evolution to match the problem requirements.

Since the system has been designed to be completely automated (i.e., without any human intervention in the nominal case), the system design shall consider the need to provide the system users (i.e., TDP Operation Centers) with the necessary information to understand the planning process, the analysis of the input requests, and, most of all, the final operation plans. In fact, different examples in the planning literature proved that a key aspect for the successful deployment of advanced planning and scheduling technologies, is the capability to deliver an *End-to-End* software system that minimizes the impact on the users' work habits while providing support to complex activities.

The remainder of the paper is organized as follows. First, a brief description of the Alphasat mission is provided followed by an introduction of the problem model. We proceed by describing the general TECO system. Then the planning approach is illustrated together with the developed algorithm. We conclude by discussing some lessons learned and possible future works.

Mission description

Alphasat, based on the new Alphabus platform, will be delivered to orbit to be operated by Inmarsat in 2013. It will carry an Inmarsat commercial communication payload and

---

[1] The reader can have a more complete picture of the state-of-the-art by checking the proceedings of the previous editions of IWPSS, the International Workshop on Planning and Scheduling for Space.

[2] http://telecom.esa.int/telecom/www/object/index.cfm?fobjectid=1138

four Technology Demonstration Payloads provided under ESA responsibility. These TDPs will be operated as secondary payloads embarked on Alphasat by Inmarsat. The four TDPs comprise:

- An advanced Laser Communication Terminal to demonstrate GEO to LEO communication links at 1064nm;
- A Q-V Band communications experiment to assess the feasibility of these bands for future commercial applications;
- An advanced Star Tracker with active pixel detector;
- An environment effects facility to monitor the GEO radiation environment and its effects on electronic components and sensors.

For each TDP a dedicated operations center (TDP-OC) will be responsible for defining and requesting the different payload experiments. The spacecraft resources (e.g., power), downlink data, and telemetry budget, and therefore experiment execution, are shared between the TDPs, while operational requests are managed at a top level by the TDP ESA Coordination Office (TECO).

The operations concept of the TDPs is broadly based around a weekly planning cycle, with data exchange over the Internet (see Fig. 1). Every week, the TDP ESA Coordination Office will provide conflict-free TDP operations requests based on:

- The available windows for TDP operations;
- The Activity request files provided by each TDP-OC.

These operations requests are assumed to be ready in time and in the proper format to be directly ingested into the Alphasat weekly schedule.

It is worth remarking that the role of Inmarsat is limited to simply executing the operations which are driven entirely by the inputs and supplementary information received from ESA (e.g. consolidated TDP operations requests and schedule) and TDP-OCs (e.g. procedure parameters). No TDP operations engineering activities are performed by Inmarsat, i.e. no TDP contingency recovery actions are defined, and no analysis of TDP performance or health are performed by Inmarsat. In other words, TDP operation requests with conflicts will simply be rejected by Inmarsat and not included in the spacecraft activity schedule.

TECO: the TDP ESA Coordination Office

As mentioned before an important role in the Alphasat Mission planning phase is played by the TDP ESA Coordination Office (TECO). This office has been set-up with two main objectives:

- TDPs Activities Coordination and Planning support – this includes: collection of activities requests from TDP-OCs; identification and resolution of possible TDP operation conflicts (w.r.t. both the platform resources and between TDPs); transmission to Inmarsat of the consolidated activity requests; reception from Inmarsat of activity execution confirmation.
- TDPs telemetry reception and archiving – this includes: reception and archiving of real-time telemetry stream; collection and archiving of the different planning files; providing historical data to the different TDP-OCs.

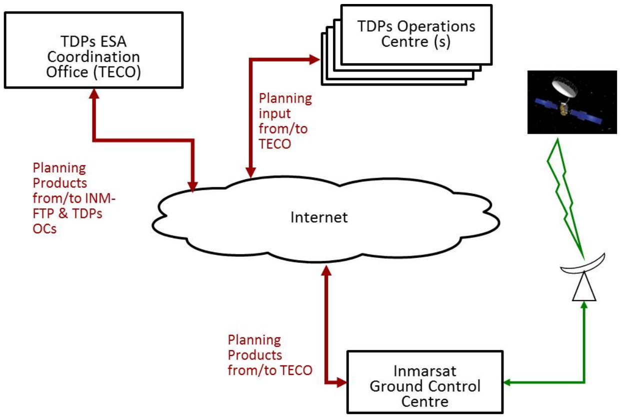


Figure 1: Overall planning interfaces

By exploiting advanced planning and scheduling technologies, the planning system has been designed to have a relatively high degree of autonomy. The goal is to have the consolidated activity plans generated automatically and all input and output retrieved automatically. Only in the case of anomaly is the TECO operator notified and required to intervene.

Mission Planning Requirements

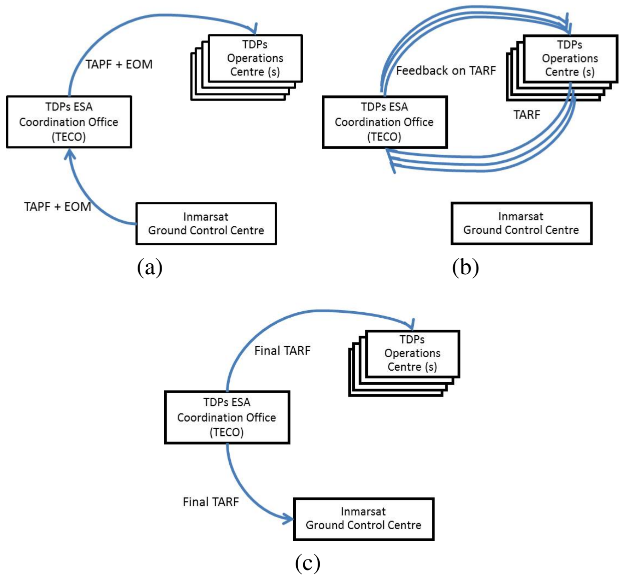
In this section we describe the Mission Planning Cycle, with a focus on the role of the TDP ESA Coordination Office. This planning cycle spans a period of a week. Each week, on day 7 by 16:00 UTC, Inmarsat makes available to TECO the TDP Activity Planning File (TAPF) containing the relevant spacecraft states and TDP operations availability windows. The different spacecraft states may have different limitations on TDP operations. A distinction can be made between those periods where TDPs could experience reduced performance (e.g. maneuvers), where no TDP related commanding activities are permitted, and where limitations exist on TDP modes.

Based on the above input, every week, on day 1 of the next cycle (by 12:00 UTC), TECO provides a TDP Activity Request File (TARF) to Inmarsat covering 7 days of TDP operations requests.

From the previous description is possible to identify the following three steps (see Fig. 2):

a. Distribution of Inmarsat input: windows availability and spacecraft status;
b. Generation of the TDP operation plan based on input requests provided by TDP-OCs;
c. Distribution of the final TDP operation plan.

The second step justifies the presence of the TECO system in general, and its planning functionality in particular. As



(a) (b) (c)

Figure 2: Planning workflow.

mentioned before, this step has been designed to be completely automated. In the next sections we present our approach underlining the planning techniques which allowed not only to produce efficient solutions but also to keep the end-users (TDP-OCs operators) in the solving loop.

Domain Model

This section introduces a timeline based representation of the problem (Chien et al. 2012).

TDP model. The central concept of the mission as described before is the presence of the different Technology Demonstration Payloads, or TDPs. For what concerns the planning problem, each TDP can be seen as a timeline which represents, at each time, the status of the payload. To represent the valid states that the TDP can assume over time, each payload can be seen as a finite state machine, i.e., a TDP can be in a finite number of states (here also named sub-modes):

- a finite, non-empty set of states, S
- a transition function $\delta : S \times S \rightarrow T, F$, which specifies for each couple S_i, S_j if the transition from S_i to S_j is allowed.

Fig. 3 shows an oriented graph representing a TDP which has five different states, the nodes. The ordered edges represent the valid transitions between two nodes/states.

Spacecraft status and opportunity windows. Another relevant aspect that has to be modeled is the spacecraft status together with the availability windows for TDP operations (that is, the information contained in the TAPF). During these windows, the Inmarsat ground segment is available to execute TDP activities. The information provided can be seen as a list of triples:

- st, the start time of the interval
- et, the end time of the interval
- $type$, the characterization of the interval. This can be one of the following:
  - No Ground Availability – during this interval, no activity can be executed from ground, i.e., no procedures (associated to any TDP activity) can be executed from the ground control center. During this interval it is however possible to have on-board procedures executed (which of course have to be uploaded in advance).
  - No TDP Activity – during this interval, no activity can be executed (neither on-ground nor on-board). The different TDPs will maintain their state through the interval.
  - TDPs off – the TDP are requested to be off during this interval. Not only no activities can be executed during this interval, but it is also required to each TDP-OC, to allocate the necessary activities to properly switch off the TDPs.
  - Maneuvers – as the different satellite maneuvers can affect some of the TDP activities, this information is provided to avoid their execution

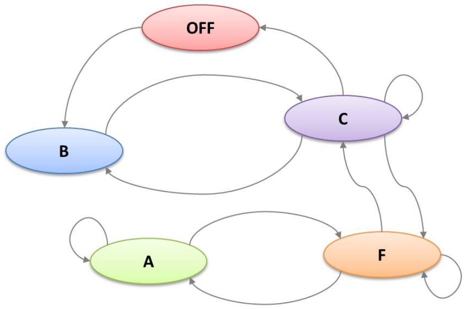


Figure 3: TDP model example.

As it is necessary to distinguish between ground control commanding and on-board only execution of tasks (see below), the spacecraft status is modeled by two timelines representing respectively the status of the ground and the spacecraft segment.

Task requests. A task request consists of a set of commands that refer to a specific payload (TDP). These commands either have to be executed from the ground control or can automatically be executed on-board without ground control intervention. As each of the commands can modify the status (sub-mode) of the payload, for what concerns our model, we associated a task to an ordered sequence, without temporal gaps, of sub-tasks (the reader can see each sub-task associated to a command or group of commands). Formally a task t_i is defined by the following set:

- A time interval $[lb_{t_i}, ub_{t_i}]$ defining the feasibility interval where the task request can be allocated;
- The associated payload or TDP, tdp_{t_i};
- A weight value, w_{t_i};

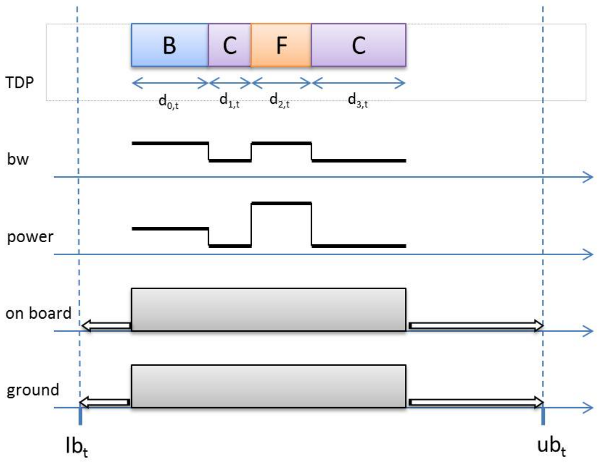


Figure 4: Task model example.

- An ordered sequence of sub-tasks $ST_{t_i} = \{st_{0,t_i}, \ldots, st_{n_{t_i},t_i}\}$, where each sub-task is defined by:
  - The duration, $d_{st_{0,t_i}}$
  - The sub-mode value, $sm_{st_{0,t_i}}$
  - The bandwidth usage, $bw_{st_{0,t_i}}$
  - The power usage, , $pw_{st_{0,t_i}}$
- A Boolean variable that indicates if a task is on-board only (value true) or if the task has to be executed both on-board and on-ground (value false) ob_{t_i}.

Fig. 4 shows a task example. It is worth considering that with respect to the timelines associated to both the ground control and the satellite status, a task can be seen as a unique entity. This does not apply to the TDP timeline; in fact, in this case it is necessary to have detailed information about the requested states of the TDP (i.e., the list of sub-modes). The same applies when resources are considered as resource consumption is also associated to TDP states.

Another aspect not represented in Fig. 4, but considered in the design of the solving approach, is that at the end of each task execution, the TDP status, as well as power and bandwidth usage, remain at the values specified from the last sub-task (for the TDP this will be C in the case of the task in Fig. 4). This aspect has to be considered, for instance, in order to have a precise estimation of resource usage.

Constraints. The final aspect that needs to be modelled is the different types of constraints that can be defined in the problem:

- Constraints between two different TDPs. This is the case in which a TDP x, in order to be in a particular status A, requires another TDP y to be in a particular status or sub-mode B, that is, $y.B$ DURING $x.A$.
- Constraints between a TDP and the status of the satellite. This is the case in which a particular status of the satellite has to be supported by specific status of the TDP. For instance in case the payloads are switched off, it is required that a TDP x is moved to the "off" status, that is, $TDPs\ off$ DURING $x.OFF$.
- Constraints between two task requests. These require that the associated tasks are either both allocated or not. The constraint can also require a minimum and/or maximum time separation between the execution of the two tasks.
- Resource Constraints. This represents the minimum/maximum availability of each resource. In the model, we consider both power and data-downlink usage.

Problem. Given the above definitions, a problem is composed by:

- A set of task requests, $Tasks$;
- A set of initial states, one for each TDP, $Init$;
- A set of constraints, $Constraints$;
- A set of time intervals representing the spacecraft availability and status, $Spacecraft$;

Solution. Given the initial problem (in particular the set of task requests), a feasible solution S consists of the set of allocated task requests, $AllTasks_S \subseteq Task$, where for each task $t_i \in AllTasks_S$ a start-time st_{t_i} is specified and all constraints as well as state variables are satisfied.

While the empty solution (i.e. $AllTasks_{S_0} = \emptyset$) is a solution, the objective is to maximize the number of allocated tasks. In particular the following weighted function shall be maximized:

$$Value(S) = \sum_{t_i \in AllTasks_S} w_{t_i}$$

Example. We conclude this section with an example to illustrate some of the aspects discussed before. For the sake of simplicity, in the example we do not consider resource constraints, and the initial states of the TDPs.

Fig. 5(a) represents a two TDPs problem with four tasks $Tasks = \{t_{a1}, t_{a2}, t_{b1}, t_{b2}\}$ (two for each TDP) and two constraints, $TDPa.W\ during\ TDPb.A$ and $t_{b1} < t_{b2}+10$. For what concerns the spacecraft availability there are two intervals in which the spacecraft is not available, the first only for on-ground commanding, the second for both on-ground and on-board execution (see the last two timelines in Fig. 5(a)). For each task the figure shows also the feasibility windows and the associated sub-task (see TDP timelines).

Fig. 5(b) shows an optimal solution to the problem. First the reader can notice that, even though t_{a1} could be allocated earlier (in fact is an on-board only task), it is allocated after t_{b1}. This is justified by the first constraint, $TDPa.W\ during\ TDPb.A$. At the end of the execution of t_{b1}, $TDPb$ remains in the status $TDPb.A$ that is required to move $TDPa$ in $TDPa.W$, and the execution of t_{a1}.

Second, as the two tasks, t_{b2} and t_{a2}, are conflicting only one is allocated: t_{b2}. In fact the latter is also linked to t_{b1} which requires that both the tasks are executed together. Therefore t_{b2} is chosen with respect to t_{a2} as it maximize the objective function. The final solution is $AllTasks = \{t_{a1}, t_{b1}, t_{b2}\}$.

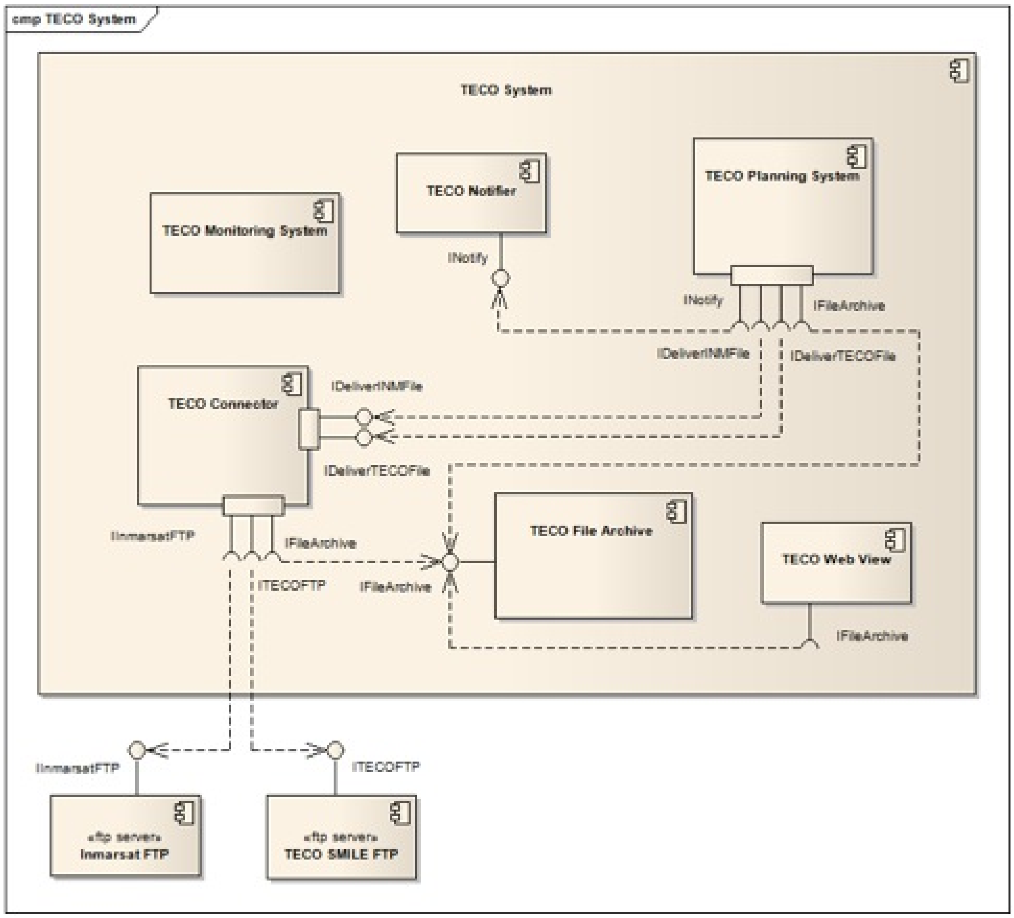


Figure 6: The TECO System High-Level Architecture.

TECO System: Architecture Overview

In moving planning and scheduling into the real world it is very important not only to produce a solution to a complex problem but also to integrate a number of features in the delivered software that contribute to creating *a complete approach to the problem* from the point of view of end-users. The high level architecture designed to implement the TECO system requirements is shown in Fig. 6, with a more detailed description of the most important components in the sections below.

TECO FTP Server and Connector. The operations concepts of the TDPs are broadly based around a weekly planning cycle, with planning data exchange over a central FTP server and direct distribution of the spacecraft telemetry over UDP broadcast to the TDP operations centers and ESA (TECO). Updates to the input parameters required by the planned activities are permitted to the TDP OCs up until their execution. The TECO FTP server serves as the mechanism for these updates to be sent by the TDP-OCs to Inmarsat, making it a crucial part of the planning cycle and requiring it to be robust. For this reason, the FTP server itself is redundant, and the system (in particular the component responsible for interfacing with external systems, the TECO Connector) has been designed so that it can cope with its failure.

TECO File Archive. The TECO File Archive is responsible for the archival of all the planning products, both input and output. It is a central part of the planning process in that the separate retrieval and planning tasks of the TECO system use the File Archive contents as the master copies of these products.

TECO Notifier. As mentioned before, one of the requirements for the TECO system is that it should be automated in the nominal case, i.e. when it is possible to conclude the planning process successfully. When this is not the case due to problems with the input files, or a software defect, it is therefore important that the TECO operators are informed in useful time so they can intervene. For this purpose, the TECO Notifier is responsible for notifying (via email) all the interested parties of meaningful events, such as the completion of the planning process or a fatal error.

TECO Web View. The TECO Web View allows the TECO operators to visualize the plan produced weekly by the system. In addition, it provides administration functionality such as log viewing and schedule inspection.

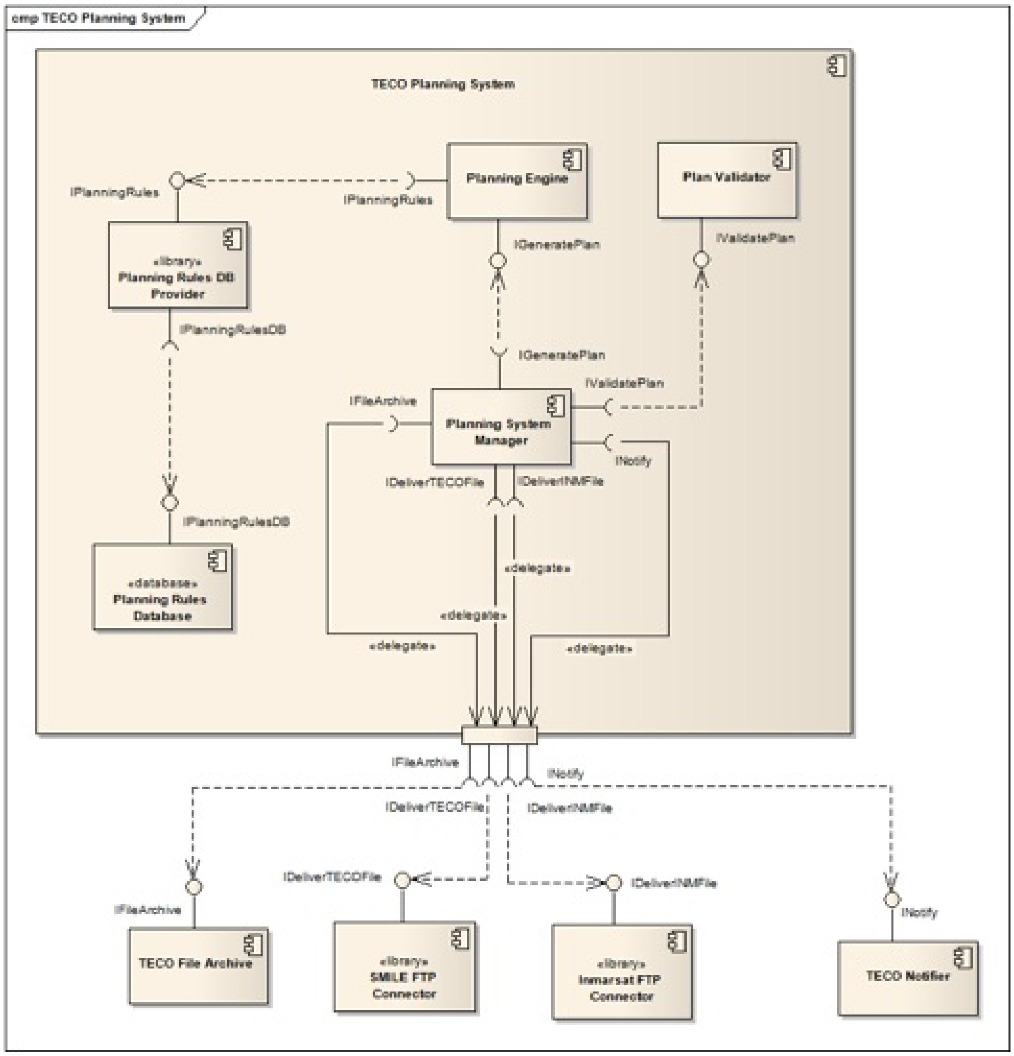


Figure 7: TECO Planner Architecture.

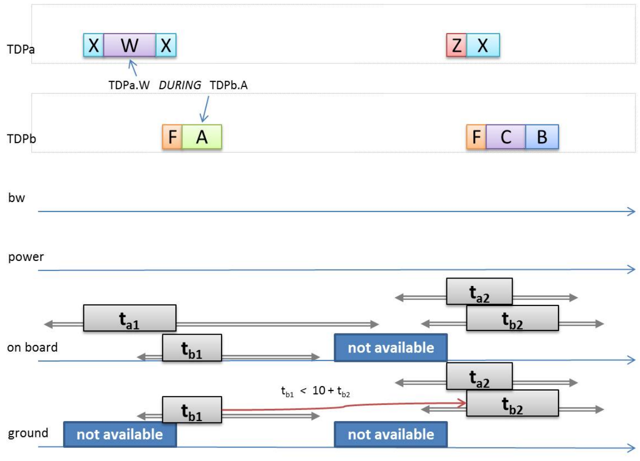
TECO Planning System. The TECO Planning System (Fig. 7) is the core of the overall TECO System. It is responsible for the generation of the weekly TDP operations schedules, and is further divided in the following components:

1. Planning System Manager: responsible for the orchestration of the individual components and for the execution workflow of the Planning System.
2. Planning Engine: based on the APSI planning and scheduling framework, this component contains the problem model and the planner. A more detailed description for this component is provided in the sections below.
3. Planning Rules Database and Provider: the Planning Rules Database contains information about the valid activities and the constraints that must be respected when generating a plan. Since these rules are stored in a database, they can be changed and are expected to evolve in the course of the mission.
4. Plan Validator: the Plan Validator is responsible for checking that the plans generated by the Planning Engine obey the constraints established by the Planning Rules and the input files for the specific planning cycle.

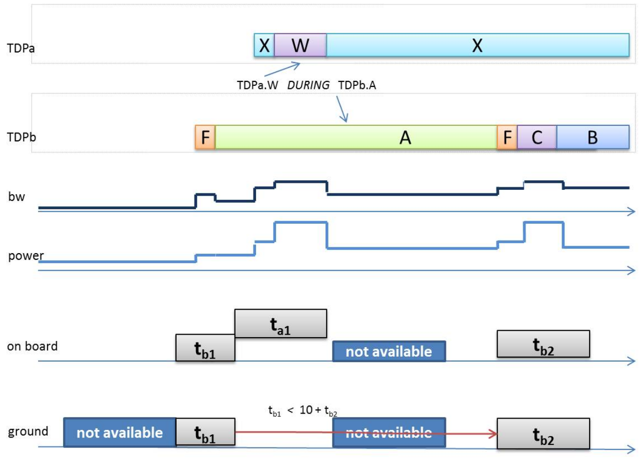
TECO Planning Engine

The core phase of the TECO workflow is the allocation of the TDP requests (Fig. 2(b)). This in fact entails different aspects such as: TDP requests analysis, exploitation of available time windows for TDP activities, platform resource limitations (e.g., TM bandwidth), interdependence between TDP modes and /or activities, TDPs operations constraints, and TDP allocation policy/priority. The TECO Planning System is responsible for this task, that is, given the TDPs operation requests as input, it produces a conflict-free plan (the TDP Activity Request File or TARF).

The TECO System design considers an iterative construction of conflict free plans; this approach gives the end-users (i.e., the TDP-OCs) the possibility, for instance, to mod-



(a) Initial requests.



(b) Solution.

Figure 5: Example.

ify their unplanned requests. The current workflow foresees three different iterations:

- TDP-OCs submit their input requests. TECO produces a plan based also on a preliminary version of the activity availability plan. Feedback is sent to the TDP-OCs in case some of their requested activities are not allocated.
- TDP-OCs submit their updated input requests. TECO produces a plan based this time on the final version of the activity availability plan (sent by Inmarsat). Also this time, feedback is sent to the TDP-OCs in case some of their requested activities are not allocated.
- TDP-OCs submit their final input requests. TECO produces a final plan and distributes it to Inmarsat (for execution) and to the TDP-OCs.

This iterative approach has been chosen in order to optimize the operation requests allocation of the different TDPs.

From a development point of view, automated planning capabilities have been exploited in the TECO Planning Engine to permit the effective use of shared resources, when the specific TDP requests contain alternative options. In this case, requests could specify:

- Alternative time intervals;
- Definition of preferred time windows (where to execute the TDP activity) contained in larger feasible time windows.

In order to achieve an advanced automated planner, the TECO planning engine has been developed as a plug-in of the APSI framework (Cesta and Fratini 2008). The current workflow foresees having a maximum time interval of 20 minutes between the reception of the input files and the sending of the plan.

APSI: the Advanced Planning and Scheduling Initiative. The APSI ESA project is the most consistent effort put into place by the Agency to create a software infrastructure to support different missions with planning and scheduling advanced technology. The project requires the development of an open software platform, called APSI-TRF, able to act as a software development environment to facilitate the application of modern Artificial Intelligence techniques for planning and scheduling to different ESA missions. During the project the software framework has been tested on a number of challenging open problems from different ESA missions (namely MARS EXPRESS, INTEGRAL, and XMM).

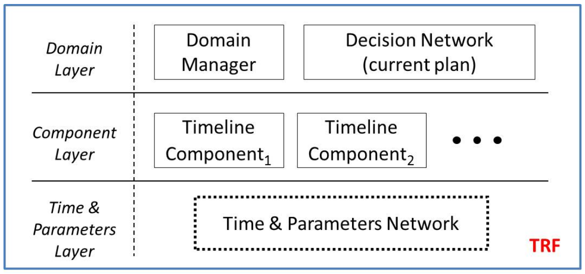


Figure 8: The general architecture of the APSI-TRF environment

The APSI-TRF design (Fig. 8) inherits from previous literature of timeline-based systems (Jonsson et al. 2000; Chien et al. 2000) and from the work in developing a general purpose planning and scheduling system called OMPS (Fratini, Pecora, and Cesta 2008). The APSI-TRF offers support to develop the domain model of different applications through different functionalities. The APSI-TRF unifies timelines of different nature under the unique concept of *component*, where each component is an entity that has a set of possible temporal evolutions over a *planning temporal horizon* – in general a component may have one or more *associated timelines*. The APSI-TRF allows representing the temporal evolution of the *components* as well as the constraints that affect their temporal evolution. Each component in the APSI-TRF is a deductive system able to proactively propagate effects of external decisions on the modeled segment of its temporal representation.

The development of the planning solution on top of the APSI framework makes the planning system easily extendable. This aspect was particularly relevant in the early phase of the mission when Alphasat operations were still under definition/design.

Solving Approach

Two solving algorithms have been considered in order to cope with the two different types of components present in the problem domain: state variable timelines (used to represent the different TDPs) and re-usable resources (i.e., ground control availability, satellite availability, power, and bandwidth usage). The two approaches have been merged into a meta-schema based on a branch-and-bound algorithm:

- At the high level, a planner allocates the different tasks on the state-variable timelines. The goal of this phase is to generate a consistent behavior for each one of the state-variables representing the TDPs.
- At the low level, a scheduler, given a solution of the planner in input, generates a feasible solution with respect to both the re-usable resources and the temporal constraints.

Algorithm 1 shows the resulting approach: here, given a problem P which has the associated set of tasks $Tasks_P$, Algorithm 1 is initialized with the queue Q = $\{P\}$ (where $Alltasks_P = Tasks_P$) and the initial best solution S_{empty} (where $Alltasks_{S_{empty}} = \emptyset$).

For what concerns the actual implementation, two algorithms have been used from the ones currently available in the APSI framework: the OMPS planner (Cesta and Fratini 2008) and the ISES scheduling algorithm (Cesta, Oddi, and Smith 2002). The planning algorithm is used, not only to check the behavioral consistency of the requested task, but also to complement these tasks to obtain continuous timelines. In fact, an activity/task has to be added between two consecutive allocated tasks in order to model the continuous usage of the resources as well as the status of the payload. These activities are considered in the scheduling phase in order to have a consistent usage of the resources. On the other side, as the set of task requests are fixed in input, the capabilities of the planner are exploited only in a limited way.

As mentioned before, during the solving process all the decisions taken are labeled with the originator of the decision and the motivation of the decision. For this reason the APSI planning and scheduling algorithms have been redesigned to return the set of unsolvable conflicts (these sets will be empty in case the solving process is successful). This information is then used to identify the next steps of the search: the branching method considers in fact not only the initial candidate solution, s_0, but also the set of unsolvable conflicts that make this candidate unfeasible. The latter is used to generate the next branches.

Engineering issues. We conclude this section by highlighting some key aspects that have driven both the design of the solving approach and the overall planning cycle.

A first aspect is how the use of planning and scheduling approaches allowed us to suggest and then introduce a change in the original workflow. In fact, the initial workflow did not foresee any iterations between TECO and the different TDPs. The short solving time permits, in the case of unresolved conflict between the TDPs specific requests, to optimize the set of task requests via an iterative process between the TDP-OCs and TECO. Even though multiple iterations can be possible, the human was also considered carefully in order to design the final planning cycle (e.g., TDP-OC operators might need time to evaluate their decisions). From this analysis a limit of three iterations was agreed. This was a compromise among software system's capabilities (e.g., solving approach), goals (e.g., TDP-OCs operators), and information availability (for instance the final satellite status timeline is only available for the last two iterations while in the first one only an estimation is used).

```
Algorithm 1: BranchNBound ($Q$, $S_{best}$)
Input: Queue of possible solutions $Q$ and current best
       solution $S_{best}$
Output: An optimal solution $S$
while $Q \neq \emptyset$ do
    $s_0 \leftarrow$ ExtractCandidateSolution ($Q$)
    if UpperBound ($s_0$)> Value ($S_{best}$) then
        $Cp \leftarrow$ planner ($s_0$)
        if $Cp = \emptyset$ then
            // a plan for $s_0$ exists
            $Cs \leftarrow$ scheduler ($s_0$)
            if $Cs = \emptyset$ then
                // a schedule for $s_0$ exists
                if Value ($s_0$)>Value ($S_{best}$) then
                    // update best solution
                    $S_{best} \leftarrow s_0$
        if $Cp \neq \emptyset$ or $Cs \neq \emptyset$ then
            $Q_{next} \leftarrow$ branching ($s_0$, $Cp$, $Cs$)
            $s_{next}$ = BranchNBound ($Q_{next}$, $S_{best}$)
            if Value ($s_{next}$)>Value ($S_{best}$) then
                // update best solution
                $S_{best} \leftarrow s_{next}$
    $Q = Q - \{s_0\}$
return $S_{best}$
```

Besides the computational capability, another characteristic of the solving algorithm suggested a re-definition of the operations. Both the planner and the scheduler have as a core a temporal representation of the problem which allows it to efficiently manage temporal aspects of a planning and scheduling problem. To exploit this capability and optimize the result of each iteration, temporal flexibility has been introduced in the task requests (instead of having fixed start-time task requests as originally designed).

Another relevant point is the explanation of planning decisions to operators: since the system has been designed to be completely automated (i.e., without any human intervention in the nominal case), the system design considers the need to provide the system users (i.e., TDP-OCs and TECO) with the necessary information to understand the planning process, the analysis of the input requests, and the final operation plans. For this reason, we include information about why a task has not been allocated at the level of the modeling of the solution. Once this information is generated by the TECO system, before distributing it, it is necessary to put it in a form that can be understood by the receivers. To cope with these problems, our current approach is based on the following points:

1. A "protocol" to provide feedback between TDP-OCs and TECO has been agreed to address differences in backgrounds among the different partners. While the TDP-OCs are experts in their specific payload , they are not required to be experts in advanced planning and scheduling technologies.
2. As described above in the knowledge representation core, all solving decisions are labeled with the solver who takes the decision and with the motivation of the decision.
3. An "Explanation Generator" module has the role of interpreting the information provided together with the solving decisions and of generating information for the system users by applying the given protocol.

It is worth remarking that a proper explanation is fundamental also to have effective iterations between TDP-OCs and TECO. Considering that the time available to the TDP-OCs to provide a new set of task requests is limited, it becomes important to provide them with the right explanation on why a task was rejected.

Lessons Learned

In this section we summarize some of the lessons learned during the design and development of the TECO system. Some of these points can also be considered as directions for future research work.

In one of our first discussions with TDP-OC representatives, we showed how an automated solver could also enable a reduction of the manpower dedicated to the planning and coordination tasks. This requirement was considered, from then on, in the project definition and design.

As mentioned before, even though at that time, the mission operations workflow was almost completely defined, we were able to convince our partners to modify the workflow and introduce iterations between TDP-OCs and TECO. The key aspect here was the efficiency of the solving algorithms together with the possibility to provide feedback to the TDP-OC operators with ad-hoc explanations of the solving process. This feedback becomes fundamental when an automated planning system is in place.

Another fundamental decision in our experience was to have a flexible architectural design of the system which allowed us to cope with the several changes experienced in the definition of the problem. This point is connected with the availability of a software framework (APSI-TRF), and its modeling ability, which enables us to both connect specialized or general solvers developed outside the framework and develop specialized interaction services. In fact, a key aspect of APSI-TRF is the presence of a flexible timeline representation module that allows exploiting alternatives in the modeling of mission features as well as developing and testing different algorithms (Cesta et al. 2009).

Something that we found missing during our experience was a Knowledge Engineering Environment for supporting the development of planning systems, which would enable a rapid prototyping approach. This type of tools allows creating a working model after a relatively short investigation by taking advantage of the speed with which this model can be implemented via the KE environment. This is fundamental to provide to the developers the basic functionalities to satisfy the model requirements and to show the end-users the main characteristics of the future system. In the current state of the art, we noticed that despite their possible role in the introduction of advanced planning and scheduling solutions to real domains, there are not many examples of these environments.

Conclusions and Future work

This paper discussed the automated planning and scheduling software system that has been designed to support the operations of the four Technology Demonstration Payloads (TDPs) that will fly on-board the Alphasat spacecraft. The TECO system will be operational in 2013 and will automatically coordinate and plan the task requests for these payloads.

At this stage the TECO system has been completely developed. During the development, the system has been intensively tested with several artificial problem benchmarks with the number of task requests ranging from few tens to hundreds (over a week time horizon). More recently the system has been validated in end-to-end test sessions with realistic task requests provided by the different TDP-OCs. Different cases have been tested, such as nominal cases, resource conflicting requests, and TDP modes inconsistent requests. These tests have shown that the TECO system can return a solution in the given time bound (20 minutes), is robust towards non-nominal cases, and can provide sufficient explanations to the TDP-OC operators.

Future work will aim at further evolving the current solving approach, in different directions. The first one consists of extending the approach to generate robust and/or flexible solutions which can better support the actual execution of the plan. For instance one of the first steps is to substitute the current scheduling algorithm with approaches able to produce flexible schedules instead of fixed-time solutions (Policella et al. 2009).

A second direction is to extend the quality of the feedback provided to the TDP-OCs operators, by adding to the current explanations possible suggestions on how to fix the unsuccessful task requests. The idea is to use in particular planning capabilities to identify reallocation of the tasks and/or new tasks to be added. This information can then be provided to the TDP-OC operators which must then validate and approve it.

From a more general viewpoint, we would like to further investigate the possibility of using "explanation" as a means to facilitate the integration of planning and scheduling solvers. In our approach, we noticed in fact as sharing adequate information among the different solvers improved the overall solving process. In future work we plan to generalize this approach and include it in the APSI framework.

References

Bensana, E.; Lemaitre, M.; and Verfaillie, G. 1999. Earth Observation Satellite Management. *Constraints: An International Journal* 4(3):293–299.

Cesta, A., and Fratini, S. 2008. The Timeline Representation Framework as a Planning and Scheduling Software Development Environment. In *PlanSIG-08. Proceedings of the 27th Workshop of the UK Planning and Scheduling Special Interest Group*.

Cesta, A.; Cortellessa, G.; Fratini, S.; Oddi, A.; and Policella, N. 2007. An innovative product for space mission planning – an *a posteriori* evaluation. In *ICAPS-07. Proceedings of the 17th International Conference on Automated Planning & Scheduling*, 57–64.

Cesta, A.; Fratini, S.; Donati, A.; Oliveira, H.; and Policella, N. 2009. Rapid Prototyping of Planning and Scheduling Tools. In *SMC-IT 2009. Proceedings of the 3rd IEEE International Conference on Space Mission Challenges for Information Technology*.

Cesta, A.; Cortellessa, G.; Fratini, S.; and Oddi, A. 2011. MrSPOCK - Steps in developing an end-to-end space application. *Computational Intelligence* 27(1):83–102.

Cesta, A.; Oddi, A.; and Smith, S. F. 2002. A Constraint-based method for Project Scheduling with Time Windows. *Journal of Heuristics* 8(1):109–136.

Chien, S.; Rabideau, G.; Knight, R.; Sherwood, R.; Engelhardt, B.; Mutz, D.; Estlin, T.; Smith, B.; Fisher, F.; Barrett, T.; Stebbins, G.; and Tran, D. 2000. ASPEN - Automating Space Mission Operations using Automated Planning and Scheduling. In *SpaceOps-00. Proceedings of the 6th International Conference on Space Operations*.

Chien, S.; Johnston, M.; Frank, J.; Giuliano, M.; Kavelaars, A.; Lenzen, C.; and Policella, N. 2012. A generalized timeline representation, services, and interface for automating space mission operations. In *Proceedings of the 12th International Conference on Space Operations, SpaceOps*. AIAA.

Fratini, S.; Pecora, F.; and Cesta, A. 2008. Unifying Planning and Scheduling as Timelines in a Component-Based Perspective. *Archives of Control Sciences* 18(2):231–271.

Jonsson, A.; Morris, P.; Muscettola, N.; Rajan, K.; and Smith, B. 2000. Planning in Interplanetary Space: Theory and Practice. In *AIPS-00. Proceedings of the Fifth Int. Conf. on Artificial Intelligence Planning and Scheduling*.

Kitching, M., and Policella, N. 2011. A Local Search Solution for the INTEGRAL Long Term Planning. In *Proceedings of the 7th International Workshop on Planning and Scheduling for Space, IWPSS11*.

Knight, R.; Rabideau, G.; Chien, S.; Engelhardt, B.; and Sherwood, R. 2001. CASPER: Space Exploration through Continuous Planning. *IEEE Intelligent Systems* 16(4):70–75.

Muscettola, N.; Nayak, P. P.; Pell, B.; and Williams, B. C. 1998. Remote agent: To boldly go where no ai system has gone before. *Artificial Intelligence* 103(1-2):5–47.

Policella, N.; Cesta, A.; Oddi; A.; and Smith, S. 2009. Solve-and-Robustify. Synthesizing Partial Order Schedules by Chaining. *Journal of Scheduling* 12(3):299–314.

Policella, N.; Oliveira, H.; and Siili, T. 2009. SKeyP: AI applied to SOHO Keyhole Operations. In *SMC-IT 2009. Proceedings of the 3rd IEEE International Conference on Space Mission Challenges for Information Technology*.

Pralet, C., and Verfaillie, G. 2009. AIMS: A Tool for Long-term Planning of the ESA INTEGRAL Mission. In *Proceedings of the 6th International Workshop on Planning and Scheduling for Space, IWPSS09*.