Planning Spacecraft Activities: An Automated Approach

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.

This paper presents an application of planning and scheduling techniques to support the upcoming Alphasat operations.[1] 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), one of the system requirements was to provide the final 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. Recent end-to-end tests have shown that the system not only can efficiently provide operational plans but it can also provide sufficient explanation of the solving process to the final users.

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 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:

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

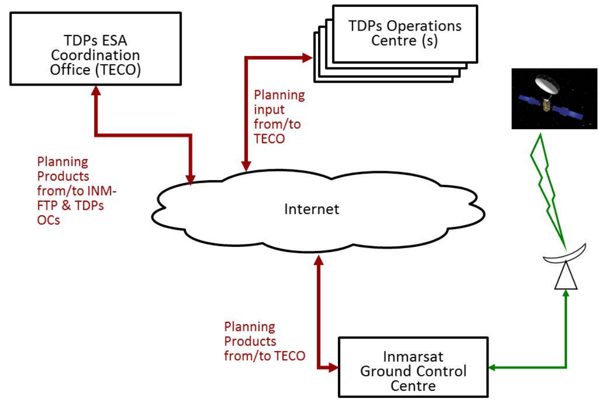


Figure 1: Overall planning interfaces

- 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.

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. In fact, the core phase of the TECO workflow is the allocation of the TDP requests. 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

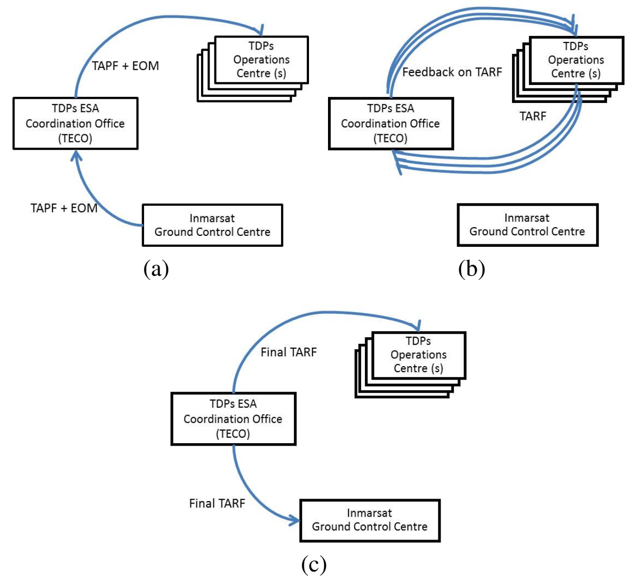


Figure 2: Planning workflow.

/or activities, TDPs operations constraints, and TDP allocation policy/priority.

In particular an iterative construction of conflict free plans (see Fig. 2(b)) is required; this approach gives the end-users (i.e., the TDP-OCs) the possibility to modify their unplanned requests. The current workflow foresees three different iterations:

- TDP-OCs submit their input requests. TECO also produces a plan based on preliminary versions of the spacecraft 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 spacecraft availability plan (sent by Inmarsat). Also at this step, 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 (that will execute it) and to the TDP-OCs.

The goal of this iterative approach is to maximize the operation requests allocation of the different TDPs, while the workflow foresees having a maximum time interval of 20 minutes between the reception of the input files and the sending of the plan.

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

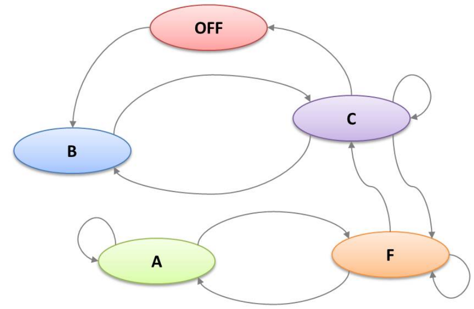


Figure 3: TDP model example.

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 opportunity windows for TDP operations. 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:
  - Spacecraft Availability – during this interval, the Inmarsat ground segment is available to execute TDP activities.
  - 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. However it is still possible to execute on-board procedures (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

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};
- 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 $x.OFF$.

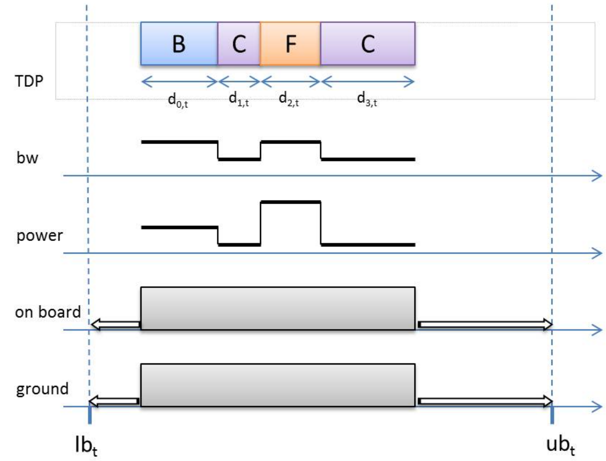


Figure 4: Task model example.

- 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, $Spacecraft$;

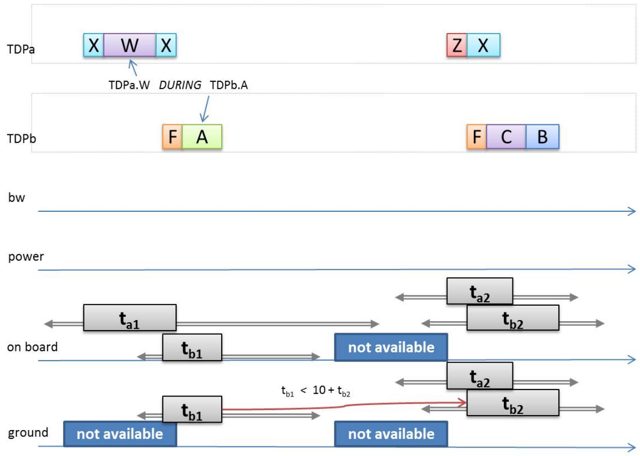
Solution. Given the initial problem (in particular the set of task requests), a feasible solution S consists of the set of allocated task requests, $AllTasks_S \subseteq Task$, where for each task $t_i \in AllTasks_S$ a start-time st_{t_i} is specified and all constraints as well as state variables are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

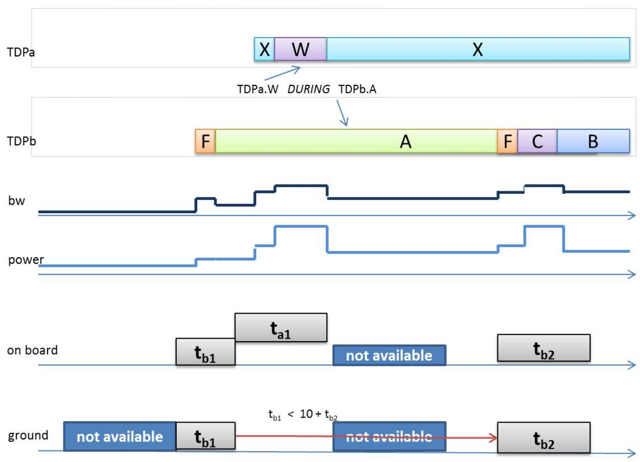
$$Value(S) = \sum_{t_i \in AllTasks_S} w_{t_i}$$

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



(a) Initial requests.



(b) Solution.

Figure 5: Example.

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 $AllTasks = \{t_{a1}, t_{b1}, t_{b2}\}$.

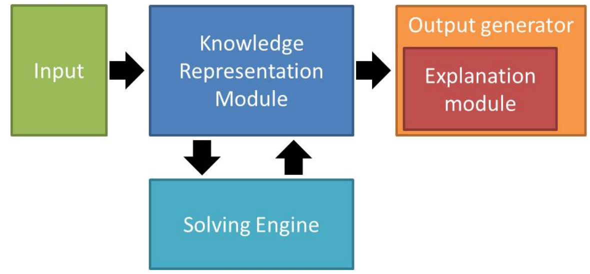


Figure 6: TECO planning engine

Planning Approach

Figure 6 shows the high-level architectural design of the TECO Planning Engine. The design is based on a common core for representing the domain knowledge. This is based on the concept of timelines, i.e., a mathematical construct that contains temporally related items. Around this core there are three different modules: an input provider, a problem solving engine, and an output generator which includes an explanation service submodule.

The input provider has the goal of populating the internal representation by importing the problem domain and the problem instance. The solving engine provides the capabilities to generate an activity plan. In particular, this can consist on a portfolio of solving libraries. Last, the output generator takes the internal representation of the final activity plan and transforms it into a requested format. This format can be for instance a specific file and/or a graphical representation of the plan. Below, we discuss a fundamental requirement of this phase, the need for complementing the final solution with sufficient information (or explanation) to let the final user (a human being or software system) understand the decisions taken during the solving process.

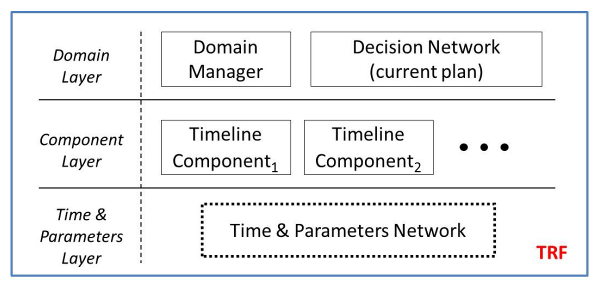


Figure 7: The general architecture of the APSI-TRF environment

The APSI framework

The TECO planning engine has been developed as a plug-in for the APSI framework (Cesta and Fratini 2008) or APSI-TRF. The framework design (Fig. 7) 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 supports the rapid prototyping of planning tools through different functionalities. The APSI-TRF unifies timelines of different nature under the unique concept of *component*, where each component is an entity with 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 the Alphasat operations concept was still under definition/design.

Knowledge Representation Module

This module allows representing the evolution of the solving process from the initial problem to the final solution(s). Following a consolidated trend in space applications (Chien et al. 2012), the representation is based on the concept of timelines, where each timeline represents the temporal evolution of one particular aspect (e.g., spacecraft utilization, ground segment utilization, and payload(s) status). The goals (that can be seen as a specific behavior of one or more timelines over a specific time interval) and the constraints (among the behaviors of two or more timelines) are used to represent the initial problem.

During the solving process different decisions are taken. Each solving decision consists of:

- a type: for instance, posted constraint, posted activity, removed goal, or removed constraint.
- a decision generator: "who" generated the decision
- a decision reason: "why" the decision was taken

The last two points have been introduced in order to coordinate different solving libraries and to support the generation of explanations. In the remainder of the section we discuss both these aspects.

Solving Approach

Two solving approaches have been considered in order to cope with the two different types of component 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). These two approaches have been merged into a meta-schema based on a branch-and-bound algorithm:

- At a 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 a 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: 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 among 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 the different 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 is fixed in input, the planner's capabilities 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, for instance, 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 makes this candidate unfeasible. The latter is used to generate the next branches.

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 ≠ ∅ do
    s_0 ← ExtractCandidateSolution (Q)
    if UpperBound (s_0)> Value (S_best) then
        Cp ← planner (s_0)
        if Cp = ∅ then
            // a plan for s_0 exists
            Cs ← scheduler (s_0)
            if Cs = ∅ then
                // a schedule for s_0 exists
                if Value (s_0)>Value (S_best) then
                    // update best solution
                    S_best ← s_0
        if Cp ≠ ∅ or Cs ≠ ∅ then
            Q_next ← 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 ← s_next
    Q = Q −{s_0 }
return S_best
```

Explanation

A relevant aspect of the output generation process is the need for providing sufficient explanation about the solving process. This is even more important in the case of TECO as the system has been designed to be completely automated, i.e., it should run without any human intervention in the nominal case.

When we talk about sufficient explanation, our objective is to provide the system users (i.e., TDP-OCs and TECO operators) with the necessary information to understand the decisions taken during the solving process and the final operation plans.

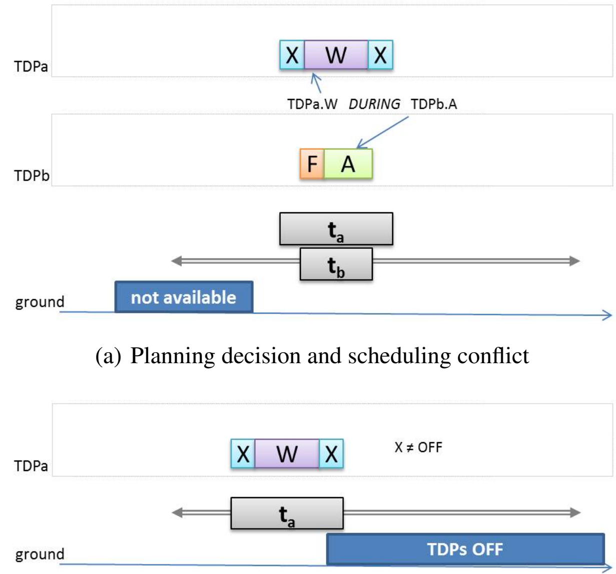
In the context of the TECO operations, a proper explanation is also needed to have effective iterations between TDP-OCs and TECO. Considering that the time available to the TDP-OCs to provide a new set of task requests is limited, it becomes fundamental to provide, for instance, the right explanation on why a task was not allocated.

To achieve these objectives, our approach is based on the following points:

1. A "protocol." This is used to exchange information between TDP-OCs and TECO. This protocol has been introduced to address differences in backgrounds among the different partners. In fact, while the TDP-OCs are experts in their specific payload , they are not required to be experts in advanced planning and scheduling technologies.
2. "Labelled decisions." All solving decisions are labeled with information about the solver who takes the decision and with the motivation of the decision.
3. An "Explanation Generator" module. The goal of this module is to interpret the information provided together with the solving decisions and of generating information for the system users by applying the given protocol.

A first aspect worthy of attention is that with this approach the final explanation is not generated directly by the different solvers. This is required as 1) the solvers can only have a limited view of the current situation and 2) to allow decoupling the set of used solvers from the final explanation generation process (and the associated protocol). Figure 8 shows two simple examples in which a task cannot be allocated. In both cases the explanation cannot be provided directly by a single solver. In the first one there are two task requests with a constraint which requires that the status W (during t_a) shall be executed while the payload TDP_b is in the status A. In this case t_b will not be allocated. In fact to satisfy the previous constraint the planner will add the solving decision $et(t_b) \leq st(t_a)$ and pass it to the scheduler. The scheduler will then fail as t_b cannot be scheduled before the start of t_a due to the presence of the "not available" constraint. In generating the explanation for the missing allocation of t_a it will then be necessary to consider not only 1) the conflict identified by the scheduler but also 2) the decision taken by the planner, and 3) the domain theory.

A second example is shown in Fig. 8(b) with only one task request t_a. In this case, while the task could be allocated in a time window, the planner will find a conflict between the final status of the task (X) and the value requested by the



(a) Planning decision and scheduling conflict

(b) Planning conflict

Figure 8: Example of explanations

platform activity "TDPs OFF." In this case the explanation process will need to consider both 1) the planner conflict and 2) the domain theory.

From the previous examples it is possible to notice that for providing correct explanation it is necessary to consider all the decisions coming from the different solvers. Even if they represent very basic cases, the examples are also useful to give an idea of the effort needed to retrieve proper information especially when considering more complex and real cases, with more activities, resources, constraints, etc.

The approach implemented to generate explanations consists in "tracing back" all the different solving decisions that are associated to a task request. This is done by exploiting the different annotations added during the solving process. Once all this information is collected, the explanation generator module identifies the specific case based both on the solvers which generate the annotations (for instance the scheduler) and on the content of the annotations (e.g., conflict with spacecraft activity).

This modular approach used for generation of explanations has also been chosen to facilitate future re-usability and evolution. For instance this can be the case of a different type of user accessing the TECO system (e.g., web-client) which can require a different protocol, or the possibility of modifying the set of solvers.

Further 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 operator 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 between the software system's capabilities (e.g., solving approach), the goals (e.g., TDP-OCs operators), and the 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).

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).

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.

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

The paper presented 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.

A further possibility would be to extend the current solving process to minimize the differences among the different solutions produced during the iteration phase (see Fig. 2(b)). Here the results obtained in the case of Minimal Perturbation problems could be re-applied (El Sakkout, Richards, and Wallace 1998).

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.

Acknowledgments

The authors would like to thank all the anonymous (and not) reviewers which provided very useful comments to the paper. A special thank goes to our colleague Simone Fratini, which supported us in the initial customization of 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.

El Sakkout, H.; Richards, T.; and Wallace, M. 1998. Minimal Perturbation in Dynamic Scheduling. In *ECAI-98. Proceedings of the 13th European Conference on Artificial Intelligence*, 504–508.

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*.