Deploying Interactive Mission Planning Tools

Experiences and Lessons Learned

Amedeo Cesta*, Gabriella Cortellessa*, Simone Fratini*, Angelo Oddi*, and Giulio Bernardi*

*ISTC-CNR, Institute of Cognitive Science and Technology, National Research Council of Italy
Via S. Martino della Battaglia, 44, I-00185 Rome, Italy
e-mail: name.surname@istc.cnr.it

Abstract

This article contains a retrospective overview of connected work performed for the European Space Agency (ESA) over a span of 10 years. We have been creating and refining an AI approach to problem solving and injected a series of deployed planning and scheduling systems which have innovated agency's mission planning practice. Goal of the paper is to identify general lessons learned and guidelines for work practice of the future. Specifically, the work dwells on issue related to some key points that have contributed to strengthen the effectiveness of our approach: the attention to domain modeling, the constraint-based algorithm synthesis, and the development of an end-to-end methodology to field applications. Desirable features of space applications useful for profitable and successful deployment on different ground segment operations are also discussed.

1 Introduction

Our research group has developed a continuous collaboration with the European Space Agency (ESA) over a span of ten years. Topic of this collaboration has been the injection of ideas from the AI planning and scheduling research area to support operations of real missions. As summarized in Figure 1 we have performed activities in different directions:

– In "product driven" activities we have developed solutions for specific mission problems. For example in a fruitful collaboration with the Mission Planning Team of the Mars-Express mission at ESOC we have addressed specific problems connected with data/command management from/to the remote spacecraft. This effort resulted in two operational prototypes currently in daily use at mission control center at ESA. In particular Mexar2 supports the downlinking problem of the Mars-Express memory [7], while Raxem [5] and its strengthened version Raxem2 [3], resolve the complementary problem of the uplink of operational commands to the probe. Such an approach inevitably entails a huge implementation effort in terms of development: specification extraction, design, implementation, maintenance.

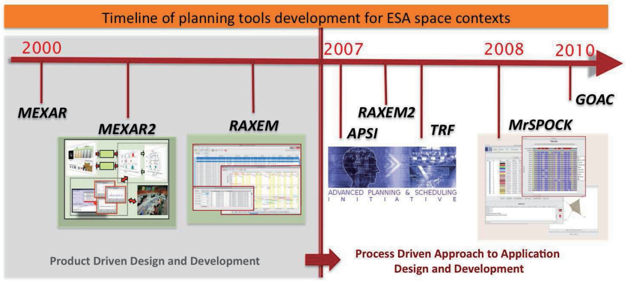

Figure 1. A summary timeline showing the work of Planning and Scheduling Team for the European Space Agency

– In "process driven" activities we have attempted at developing general purpose tools for facilitating the design and synthesis of new products. In particular within a larger project that spanned from 2007 to 2009, called the Advanced Planning and Scheduling Initiative (APSI) we have developed the Timeline-base Representation Framework (Apsi-Trf) [9]. APSI aim was to develop a general framework to improve the cost-effectiveness and flexibility of the mission planning systems (MPS) development. The tool offers a Java plaform with primitives to capture the specificity of an application domain and a given problem, thus fostering rapid and fast prototyping. An example of such support is represented by the MrSPOCK a operational product for Mars-Express long term planning. A new example of "process driven" approach is the GOAC project an ongoing activity for designing future generation of robotic controllers [20].

This paper aims at presenting a comprehensive picture on these activities and an attempt at drawing general points that emerge from the work. In revising our collaboration with ESA we focus also on the open challenges and lessons learned. An inspiring paper has been [12] that contains a similar effort at gathering a number of lessons learned while deploying automated planning and scheduling systems for space applications at the Jet Propulsion Laboratory. Similarly to that work, we here report on our experience and try to give an updated vision of the open challenges and achievements of AI based solutions for space contexts. One specific aim we have is to highlight our steps toward the consolidation of a robust methodology for fast development of ground segment mission planning support tools, showing the transition from a "product driven" approach to a more general and structured way of working, "process driven", based on a general open framework.

Plan of the paper. In the following Section 2 a brief introduction of the single activities creates the context for the critical discussion. The subsequent Section 3 presents the lessons learned according to two main dimensions: the first dimension suggests some more general challenges and issues that are recurrent when designing applications for space contexts; the second dimension addresses a more pragmatic aspect which is related to the ideal characteristics that a planning and scheduling application should possess. Some conclusive remarks end the paper.

2 Deployed Work

Aim of this section is to provide the context for the debate on the various challenges and lessons learned during the development of planning and scheduling applications within the ESA space environment. More in particular it illustrates the sequence of systems developed for ESA by the Planning and Scheduling Team at ISTC-CNR.

The opportunity for a fertile collaboration with ESA has been given by the MARS-EXPRESS[1] interplanetary mission, launched by ESA in June 2003 and prolonged until 31 December 2012. The MARS-EXPRESS probe has created the opportunity for providing spectacular images of Red Planet and other valuable data such as high-resolution photo-geology, mineralogical mapping and mapping of the atmospheric composition, study the subsurface structure, the global atmospheric circulation as well as the interaction between the atmosphere and the subsurface, and the atmosphere and the interplanetary medium.

The mission has been the first of an ESA program for using a rapid prototyping design and that implemented concepts of re-using of modules between different missions to shorten both development time and costs. Given the relatively short design time the actual operational phase presented some problems that were seriously able to prevent a full exploitation of the mission activities. For example the memory dump problem had been underestimated before launch and actually impacted on the return of scientific data. In this scenario, our work has been commissioned by ESA to solve different subproblems identified within the mission planning management. This effort led to the development of several prototypes that quickly were turned in operational modules given their effectiveness. They are now operating in various parts of the mission planning cycle.

Indeed the success of the first MARS-EXPRESS planning products has opened the possibility within the Agency for more general initiative dedicated to improve the synthesis of more innovative products – hence the APSI project for example.

2.1 MEXAR and MEXAR2

MEXAR2 is an advanced software tool based on the interactive problem solving architecture deployed to ESA-ESOC during 2004 and which is operational on a daily basis since beginning of 2005. MEXAR2 is devoted to the support of data dumping activities of MARS-EXPRESS, and it is an operational version of a prototype named MEXAR, synthesized during a previous study, 2001 to mid 2002.

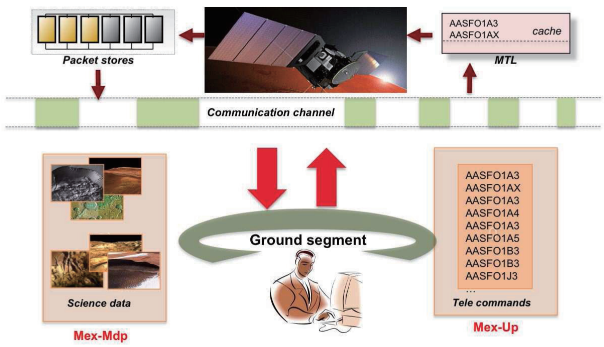

Figure 2. The MARS-EXPRESS memory downlink and command uplink problems

The MEX-MDP problem and its criticalities. The MEX-MDP problem is an important aspect within the overall process needed to manage science data return (see Figure 2, left part). The space-probe continuously produces a large amount of data resulting from the activities of its payloads and from on-board device monitoring and verification tasks. All these data are to be transferred to Earth during bounded downlink sessions. On board data generally require to be first stored in an on board memory and then transferred to Earth. The on-board memory is subdivided into different banks, called *packet stores*, in which both scientific (*science* from payloads) and spacecraft management data (*housekeeping*) can be stored. Each packet store

[1] http://www.esa.int/SPECIALS/Mars_Express

assigned to science data is managed cyclically, so in case new data are produced before the previous are dumped to Earth, older data are overwritten and the related observation experiments have to be re-scheduled.

The main problem to be solved consists in synthesizing sequences of spacecraft operations (dump plans) that are necessary to deliver the content of the on-board memory during the available downlink windows.

The most important criticality of the problem was the *uncertainty* in the data production. In particular, due to differences in compression algorithms the amount of data produced by some of the activities are not exactly predictable in advance creating once in a while the need to quickly recompute the short term plan before the commands uplink.

The previous management of the problem ended up being very time consuming because mission planners were essentially proceeding by "Trial and Error" on a textual plan. In addition, the activity of "manual" refinement was possible having a reference period of one or two days but was cognitively very difficult on a larger horizon. Overall this planning process too often produced dump plans not sufficiently reliable with respect to *overwriting*, and also entailed a significant amount of time spent on repetitive and tedious low level tasks.

The MEXAR and MEXAR2 solution. Three years before operations, authors have had the opportunity to study the memory dumping problem of MARS-EXPRESS. This previous experience produced MEXAR [8], a demonstrator able to solve a core problem close to the current one and started a trustful interaction with the mission planners. The prototype was never tested on real data because not available at that time, but different algorithms were developed based on a CSP approach [29]. The daily dump plan synthesis at ESA was so critic, and time consuming for humans, that in June 2004 we received a detailed description of the real data format and have been entrusted with the task of possibly improve the current work practice.

Since February 2005 an interactive AI tool, named MEXAR2 is entered in the daily loop of mission practice. An AI based solver finds an initial solution and allows users to influence the solving process by manipulating "modifiers" (input parameters), and preserves users' responsibility on the choices [7]. A main advancement for the solver has been to solve multiple mission days problems while the interaction part offers to the mission planners a set of additional statistics and information on the current solution and the possibility to obtain and compare different solutions. MEXAR2 has significantly increased the science data return and reduced the mission planning team's workload by half, compared to the previous method. Science data are generally available on Earth as early as possible and overwrites of the science packet stores are minimized. The tool has also contributed to reduce the mission costs thanks to a better managements of the ground stations reservations [4].

2.2 RAXEM and RAXEM2

RAXEM and RAXEM2 are subsequent versions of a software tool deployed to ESA-ESOC for the continuos support to commands uplink activities for MARS-EXPRESS designed to help mission planners in managing the uplink of tele-commands (see Figure 2, right part).

The MEX-UP problem and its criticalities. The MARS-EXPRESS spacecraft is not able to plan and execute science operations in a fully autonomous way, hence its plans arrive from the ground on a continuous basis. A particular on-board memory block, called Mission TimeLine (MTL), is replenished by uploading telecommands (TCs) from the ground (see Figure 2, right part).

The spacecraft activities for each month are determined in accordance with the Medium Term Plan (MTP) for the concerned period (typically 4 weeks). Based on the given MTP various *operations requests* (OR) are generated. During the daily planning activity the operations requests are converted into MTL *Detailed Agenda Files* (MDAF) by the Mission Planning System: the MDAF expands these operations into time-tagged tele-commands which are transferred to the spacecraft. On-board the spacecraft the TCs reside in the MTL buffer ordered by execution time. At the specified execution time, each TC will be released and removed from the MTL. Two constraints make this problem hard: the limited bandwidth of the transmission channel and the finite capacity of the on-board memory (MTL) where the commands have to be stored, waiting for the execution.

The MEX-UP *problem consists in producing an* uploading plan *for the set of* MDAFs*, considering the available uplink windows, the status of the* MTL*, and the priority of each* MDAF.

Some critical aspects of the problem have revealed the necessity of an intelligent support for the uplink management: (a) uplink is an incremental process so there is a need to *dynamically manage a current plan*; (b) there is always need to insert additional command sets to upload for some emergency or *unforeseen events*; finally (c) planning uplink is a continuous task that, even being very important for the mission, tends to become routine hence inevitably prone to errors.

RAXEM and RAXEM2 solution. Both RAXEM and RAXEM2 use an AI constraint-based modeling and solving approach to plan each command file for uplink, retaining a backup window wherever possible, keeping the on-board timeline as full as feasible, and ensuring the safety of the spacecraft at all times. The solver employs a constructive algorithm which provides a good trade-off between speed and optimality of the solution [5, 3]. The user can provide hints to the solver. Planning parameters can be altered for individual MDAFS. The possibility of tuning the solution is provided allowing to perform what-if analysis. A key point of both RAXEM and RAXEM2 tools is to support the continuity of work of mission planners. They are in continuous contact with payload PIs and may receive commands to be uplinked distributed over time including the possibility of having to accommodate new activities in a short notice. As a consequence tools have been endowed with an interaction layer that supports incremental plan definition and management (see the specific discussion on plan management in [5].

In addition, in RAXEM2 a further module has been added that better supports the complete and continuous management of the uplink problem. Previous work practice in fact turned out to be highly fragmented: RAXEM was part of a loop which entailed the use of an external semi-manual process in order to produce the *Spacon Instruction Form* (SIF) needed to the operator to execute the uplink plan and to maintain an updated detailed information of the on-board memory dedicated to commands status. This led to the need of having a tool to better embrace the whole work cycle in a more organic and rational way. The new RAXEM2 generates SIFS and populates a database which records the uplink history, maintaining also track of users' responsibilities and guaranteeing a continuous complete control on the uplink steps.

Thanks to RAXEM2, engineering workload has been reduced from five hours to below one hour. The software was up to expectations with respect to plan generation handling of emergency solution feasibility. Plan management has proved to be a very valuable feature to minimize human errors during continuous operations.

2.3 The APSI project and the APSI-TRF

Both the above described projects are examples of great success in introducing AI techniques within ESA mission planning contexts, and have shown clear advantages in term of performances and users' satisfaction. However, it is worth highlighting how a great effort and amount of time has been necessary to both understand the problems, capturing all the specificities, and to create a model of the relevant aspects of the domains and the problems themselves. The work done for MEXAR2 has been in some way useful for the RAXEM2 tool, but in general the development process has been time-consuming and extremely demanding. In addition both MEXAR2 and RAXEM2 were devoted to solve very specific and isolated problems, while the space missions offer many opportunities for relying on AI P&S to solve problems. This experience suggested us to operate in a more systematic way trying to identifying commonalities and similarities among the different domains and problems within the space contexts, with the aim of developing a more general and flexible approach that can be applied to different cases. The general pursued idea is the one of improving the "process" of tool development taking advantage of the state of the art planning and scheduling technology.

The opportunity to investigate in this direction has been provided by the APSI (Advanced Planning and Scheduling Initiative) project, an ESA initiative to develop an open framework for the flexible support of mission planning systems. The result of this project has been a quite general software framework, named APSI-TRF (where APSI-TRF stands for Timeline-based Representation Framework), which provides the basic elements for modeling the relevant entities in the space contexts [9]. In these contexts the relevant aspects are represented by the ability to deal with *time*, *resources*, description of *operational modes*, and *synchronizations* among events. The APSI-TRF offers a structured library for managing effectively and efficiently these elements and provides the flexibilities to model different domains and problems. It is centered on the concept of *timelines* which describe features that evolve over time, a concept that is particularly suitable and close to the way of working of human mission planners, thus offering also a good metaphor for managing the interaction with the users.

The APSI-TRF has been shown at work, for the modeling and resolution of a problem identified within the MARS-EXPRESS program. In particular the APSI-TRF has been used by our team to deploy an application, named MrSPOCK, described below, that is devoted to the support of MARS-EXPRESS Long Term Plan synthesis.

2.4 MrSPOCK

MrSPOCK is an application developed to support the collaborative problem solving process between the science team and the operation team of the MARS-EXPRESS mission.

The MEX-LTP problem and its criticalities. The two groups of human planners mentioned above iteratively refine a plan containing all activities for the mission. The process starts at the long term plan (LTP) level – three months of planning horizon – and is gradually refined to obtain fully instantiated activities at short term plan (STP) level – one week of planning horizon. This process continuously leads to weekly STPs, which are then further refined every two days to produce final executable plans.

Goal of MrSPOCK has been to develop a pre-planning optimization tool for spacecraft operations planning and specifically we have focused on the generation of a pre-optimized skeleton LTP which will then be subject to cooperative science team and/operation team refinement (see [6] for a more detailed description of the whole work).

A critical point in developing an application to produce the Mars-Express skeleton LTP is the consideration to be given to a great number of operational constraints that cannot be removed after four years of daily mission operation practice. In order to capture the work practice we had to cope with very specific constraints that are difficult for the general purpose solving framework but more easily to be taken into account in a domain specific solver.

The MrSPOCK solution. MrSPOCK, the "Mars-Express Science Plan Opportunities Coordination Kit", is a new tool which combines diversified research aspects from the planning and scheduling area. It has been developed on top of the Apsi-Trf. The leverage offered by the software development platform has allowed us to deliver increasingly accurate and performant versions of the complete system on May, August 2008 and January 2009. During 2009 and 2010 the system has entered a long phase of testing and comparison with current support tools. It will be the main tool for Long Term Mission Planning for Mars-Express starting January 2011.

An interesting aspect of the system is the hybrid combination of a core constraint-based representation that supports timeline-based planning and scheduling, with an optimization algorithm that exploits such representation and an interaction front end which has multiple features.

Apart the fielded application it is worth highlighting here the interesting leverage we obtained with respect to our previous experience in ESA projects, e.g., [4], due to the use of the Apsi-Trf. This general framework has allowed us to capture an amount of constraints with a basic domain description language. In our previous experience described in the Mexar2 tool we have used a model-based representation based on timelines and several principles of mixed-initiative planning that are research products of our area, the whole implementation was done on-purpose for the application. In MrSPOCK the amount of the general purpose modules used in the implemented system is quite high with respect to our previous work. It is also worth mentioning that the development of a solver entirely based on domain independent solver would require the customization of an amount of specific knowledge in the domain description with a consequent production of a rather cumbersome domain model. Our choice has been to use Apsi-Trf for clean modeling purposes while relying on a specific module for driving an efficient problem solving.

3 Lessons Learned

The presentation of lessons learned and challenges we encountered during our work within space missions focuses on two dimensions that have a different level of granularity. In particular, there are some challenges and critical issues that are general, recurring and common to space domains, while others may be interpreted rather as desired features and functionalities that the systems should possess in order to effectively support the planning and scheduling within real contexts. In this section the two different perspectives are introduced.

3.1 General challenges

The general issues that must be taken into account when developing applications within space context can be grouped into three main topics: (a) the *modeling approach* used to represent the domain of interest and addressed problems, (b) the type of available *algorithmic support*, (c) the overall *development approach* which ensures a support for the entire life cycle of the problem solving. In this section, these three aspects are examined in light of our experience within the work introduced before.

3.1.1 Modeling approach

In developing mission planning tools two issues that have been proven being critical are domain-knowledge capture and plan presentation and maintenance. The two aspects are strictly connected: the closer is the modeling paradigm (used to represent problems, goals and domain theories) to the plan and solution representation the simpler and cost-effective are the domain-knowledge acquisition (on one side) and the plan maintenance and verification (on the other side). As far as domain-knowledge representation is concerned, the closer to the practice of the domain experts (human planners) are the modeling primitives available in the planning system, the simpler and safer (in the sense of correctness) will be the domain knowledge acquisition from domain experts. As for the plan and solution presentation, the closer to the existing practice is the plan database representation, the easier and more effective will be the interaction of the users with the system. Allowing users to understand plans is a necessary precondition to enable a productive interaction with the system.

Hence the idea of using existing practice as a driving factor for choosing the planning paradigm and the plan database representation primitives. The "action based" approach [17, 18], widely used in the planning research community, models the planning problem in terms of the actions that an agent can perform to modify the given world (represented as a collection of action schemata that describe pre-conditions for action application and the effects of the actions on the world). The world is usually described in terms of static properties that can be true or false. It is not so straightforward to reconcile this approach with the current practice in mission planning. This is because:

1. The world in described in terms of "events" that happens in time instead of in terms of static properties that can be true or false
2. The working domain is constituted by physical and natural systems interacting with each other instead of being constituted by a single agent that interacts with the world
3. These systems are described in terms of "states" they can assume (and consequent constraints on them often are represented as Finite State Automata) instead of in terms of "what they can do and when"
4. The interactions of these systems are described as patterns of temporal synchronizations between the states they can go through instead of in terms of effects that they produce in the surrounding world
5. The world presents non trivial limited availability of some resources, and the interactions of the systems must take into account the competition for limited resources
6. The problems are described in terms of system's desired behaviors and resource optimization instead of being formulated in terms of "which actions will achieve a given status of the world's static properties?"

These are just some examples of the distance between the classical planning paradigms and the problem that the planning technology is required to address to be deployed in mission planning. In fact the action based paradigm presents a number of downsides. Most of them stem in the ground assumption according to which the executive agent performs actions against a world of a-temporal properties.

An approach closer to sketched work practice is the so-called "timeline-based planning and scheduling" [27, 19, 9]. According to this approach the planning domain is described in terms of temporal evolutions, called timelines, of sub-systems, and their interactions are modeled as temporal synchronizations between their behaviors. This paradigm is able to capture plans and planning problems in mission planning because:

1. The timeline is a primitive of representation natural in mission planning (as well as interactions between systems described in terms of temporal synchronizations among timelines)
2. Timed events and temporal reasoning are embedded in this approach (since the world is described by subsystems temporal evolutions)
3. Scheduling and resource reasoning can be efficiently applied to timeline-based domains ([9])
4. Uncontrollable temporal events (like flight dynamics, visibility windows, exogenous events and so on) can be naturally represented as timelines
5. The problems that can be addressed concern the achievement of a desired behavior (for one or more subsystems) compliant with the temporal synchronizations and resource allocations that describes correct sub-systems interactions.
6. Planning problems, surrounding conditions and planning solutions are all represented by means of timelines. Hence human planners can easily understand and interact with what they get from the system.

For these reasons a timeline-based approach simplifies the elicitation of the domain expert knowledge, the problem formalization and the plan database knowledge representation.

3.1.2 Algorithm support

A central role in the *end-to-end* development approach for mission planning tools is played by core algorithms. Mission planning problems can be often modeled as integrated planning and scheduling problems. They involves several complex constraints and many conflicting objectives that users want to accommodate. According to our experience, a rigid algorithm that finds a specific solution is seldom the successful choice. On the contrary, a multi-objective formulation of the problem, connected to a heuristic approach able to accommodate divergent goals and integrating user's knowledge, can be the winning choice. The analysis that follows focusses on three different issues for developing interactive mission planning tools: (1) efficient *batch algorithms*; (2) procedures for supporting the user *interaction* process; (3) integrating *planning* and *execution*.

In general, our approach for solving planning and scheduling problems combines heuristic search and constraint reasoning. Specifically, according to our experience, heuristic search in combination with constraint reasoning techniques, has emerged as a robust methodology to quickly produce good-quality solutions for a variety of mission planning problems. Despite the advantages of finding *optimal* solutions to scheduling and planning problems, we preferred to use heuristic and incomplete approaches over complete and systematic methods. This is for several reasons. First, generally a complete method cannot scale on large instances. Second, in some cases reaching optimal solutions is meaningless, as in practice we are often dealing with models that are simplifications of the working domain (e.g., this is the case of long-term planning of satellite activities). Third, according to our experience, the use of heuristic strategies allows a smoother integration of the mission planner choices in the search process.

Batch algorithms for mission planning. As quickly mentioned in the previous sections, mission planning is a large process involving different stages and the interaction among several teams (e.g., Science Working Teams, Mission Analysis Team, Flight Dynamics and Flight Control Teams). In addition, we can identify at least three phases for planning: long, medium and short term planning, which respectively have durations of several months, one months (or less) and one week (or less). Along these phases planning activities are *incrementally* refined from an horizon of several months to one week or less. In fact, as some activities cannot be predicted one week in advance (e.g., flight dynamic requests), a very short-term plan can be generated each one or two days. The use of batch algorithms is the first step in developing a solution to a mission planning problem, such that, given a set of input parameters, a solution is generated over a single run of the algorithm. No user interaction is allowed during the resolution process and there is the hypothesis that no changes or additions to the problem constraints are allowed at *execution time*. These last two important issues (user interaction and execution) are considered in the next two subsections.

Hence, the horizon of mission planning problems can span from several months to one or two days. In the first case, the output of the planning process is generally a *skeleton plan* (an abstract plan) used as a baseline for future activities and for the coordination of the involved working teams. This is the case of our APSI-based application, MrSPOCK, [6], where the goal is the generation of a pre-optimized skeleton plan containing three types of activities corresponding to three different orbit phases around *Mars*: (1) time interval around the *pericenter* (the closest orbital point to the planet); (2) time interval around the *apocenter* (the farthest orbital point from the planet); (3) time interval *between* the pericenter and apocenter passages. The optimization problem is a multi-objective one, where we accommodate three different objectives: the number of opportunities for doing science (performed around pericentre passages); the opportunities for downlinking data to Earth (mainly performed in the rest of the orbit, except in the case that *maintenance* operations are performed around apocenter passages) and the uniform distribution of uplink windows having a given minimal duration (generally 4 hours). As mentioned we have proposed a heuristic algorithm able to generate solutions over an horizon of several months. Our procedure is based on a genetic algorithm for multi-objective optimization, using a core constraint-based algorithm for generating (and decode) each generated chromosome. This algorithm is an effective way to generate a set of no-dominated solutions – a solution S_1 dominates S_2 when each component of the S_1 objective function is better than the corresponding component in S_2 – close to the set of Pareto optimal ones. It allows mission planners to choose among a set of potential good solutions - each one represents a different accommodation of the objectives - which are equivalent from a point of view of the optimization model, but can be very different from the point of view of an experienced mission planner. Hence, solutions to long-term mission planning problems are approximate solutions, characterized by the lack of many kinds of information.

On the contrary at short-term (or very short-term) level the input planning problem contains information (in some cases) for generating as output the full sequence of commands for controlling a satellite. This is the case of the *memory dumping problem* solved by MEXAR2, for instance. This problem contains many kinds of constraints, which contrast each other. For this reason, also in this case the problem is a multi-objective one, with four possible components: the volume of data lost; the *robustness* [28] of the solution; the weighted average delivery time of the data (takes into account priorities); and the size of the output commanding plans. A mission planner have to find a good compromise among these contrasting objectives and in some critical situations the problem is oversubscribed. Our heuristic algorithm is based on a core max-flow algorithm allowing (the tool is still operative) the mission planners the control of the four components of the objective function. Another heuristic algorithm has been proposed in [34], where the authors define an LP-based algorithm for solving a simplified version of the memory dumping problem. In particular, the authors make the hypothesis that data acquisition of telemetry data in the on-board memory can only happen during no-communication windows. Hence, during no-communication windows data volumes in the set of on-board packet-stores (there is an independent packet store for each instrument) can only increase. Whereas, data volumes can only decrease during communication windows. As a consequence, for each communication window, data downlink operations from the set of packet stores can be executed in a single step (no pre-emption is needed), with a consequent decrease of the size (total number of downlink commands) of the generated downlink plans.

Integrating User Knowledge. Since the first version of the MEXAR software [29, 8] we started a research activity with the main goal of defining a decision support system (DSS) for solving mission planning problems. The main idea was to integrate human strategic capabilities and automatic problem solving algorithms to find solutions accommodating different and contrasting goals, under the full control of the mission planners. In fact, despite the use of complex models and powerful algorithms, it still remains a *gap* between the user desiderata and the set of found solutions. Integrating user knowledge in the search process helps to fill this gap. Over the years we analyzed and applied several techniques to include user's choices, both at strategic and detailed level, from a point of view of the algorithms we can consider the following two modalities of interaction.

What-if analysis can be an effective way to obtain alternative solutions, which takes into account priorities, preferences and user's experience. Under this definition there is the simple and powerful idea of comparing different solutions operating modifications at different level of granularity: from the modification of the position of single activity (e.g., change the scheduled time of a scientific observation) to the use of different solving strategy for the whole problem. The use of *incremental algorithms* can improve the efficiency of this kind of analysis. For example, if the modifications are operated only on the temporal constraint and the reference model is the Simple Temporal Problem, then are available efficient algorithms for operating incremental modifications [10, 30] on the reference solution. In general, due to the complex constraints involved in mission planning problems, even a small modification can induce high computational efforts to regain a feasible situation. *Explanation-based reasoning* can be used to deduce the *culprit* of a failure, as well a subset of constraints justifying an action of the solver (e.g., a value removal or a bound update). An example of this kind of application is described in [35], in general *explanation* [21, 1, 22] is a promising research area with increasing applications in our future research work.

Human-Guided Search (HuGS) can be an alternative and effective way for integrating user's knowledge. The basic idea [2] is to allow an user to guide the exploration of the search space. This is a very promising direction of our future research work. Within the literature we consider a valid support the work [23], which present a survey of techniques targeted to investigate interactive optimization. Another work is [15], which presents a framework for the visualization of optimization algorithms - in particular constraint-based local search (CBLS) algorithms - in order to identify their bottlenecks or pathological behaviors, and suggest remedial techniques.

Planning and Execution. One of the challenges for mission planning systems is the generation of plans robust to uncertainty and adaptable to change. In fact, in many practical situations, the effective value of many variables of the plan are only known at execution time and their values are not under the control of the mission planners. Hence, even if an initial reference plan can be created via a batch algorithm, its adaptation, which takes into account the real values of the plan variables, can be only obtained via a *dynamic planning* approach, where a reference plan is *continuously* updated on the basis of the performed observations on the real word. As suggested in [12], a general dynamic planning approach applies incremental planning techniques in conjunction with a hierarchical approach. In the latter the long-term planning horizon is planned only at a very abstract level, whereas shorter and shorter planning horizons should be planned in greater detail. The devised general framework in [12] posts several algorithmic challenges and many of them represents open research areas. One challenge, especially at short-term level, is to guarantee that the generated plan is *controllable*, that is there is a strategy for executing all the the planned activities under the user's control that satisfies all requirements, in all situations involving the uncontrolled events. In the case the source of uncertainty are only the temporal constraints and the adopted model is the Simple Temporal Problem an interesting property is the so-called *dynamic controllability* [26, 25], where it is assumed that each uncertain duration becomes known (is observed) after it has finished, and the property requires a successful strategy that depends only on the past outcomes. About the verification of dynamic controllability the work [11] proposes a model-checking verification tool suitable for verifying flexible temporal plans. Another proposal, based on the idea of compiling alternatives in the plan, is proposed is [14]. When resource constraints are also involved in the plans, steps towards plan controllability are proposed in [33, 31].

3.1.3 The approach to application development

The development experience gained in these years working for ESA has also brought a contribution in relation to the synthesis of an approach to the process of system development. In particular, two concepts have emerged from the work of design and development: (a) End-to-end development; (b) Product driven vs. process driven approaches.

End-to-end development. According to our experience the key for successful deployment of AI planning and scheduling technology is the ability to deliver an *End-to-End* software system that minimizes the impact on the users' work habits while offering a real support to regular difficult activities. As an example, a seamless connection of the delivered software to the actual work environment is an additional requirement that is always to be taken into account. To foster a smooth deployment of our solutions within the working environment we have developed an approach during the years that is depicted in the simple drawing of Figure 3. As shown in the figure, three main blocks integrate their work: (a) a domain model able to capture the segment of the real world relevant for the problem to be solved; (b) a solver which addresses the problem and forces either search decisions or solution representation on the domain model; (c) a set of user-oriented interaction services that allow end-users to access the domain representation, the solution representation and, usually, a set of parameters associated with the problem solver.

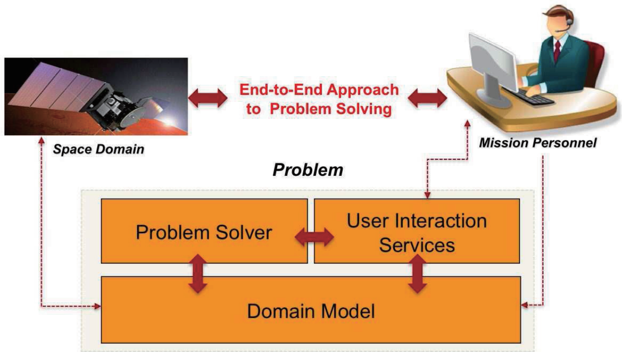

Figure 3. Supporting the development of an end-to-end approach to problem solving

An effective domain model is the core of the problem solving approach based on AI techniques. The domain model is used not only to define the problem but also to create a data-structure for representing a model of the solution that is used before, during and after the problem solving phase to create a "live" representation of the domain continuously available for different purposes. Implicit in the drawing is the schema of problem solving as continuous work on a current solution representation. The interaction layer closes the loop with the end user and allows the creation of "closed applications" that are usable in the operative environment.

Product driven vs. process driven approaches. As already mentioned, early development efforts for the development of space applications have been characterized by a significant work to understand, represent and model the specificities of both domain and problems. This involved an extremely long development time, also producing solutions not particularly general and extensible. Starting from there, and demonstrated the usefulness of the approach, the idea has been to move toward a different approach to development that is more general and able to guarantee more flexible and reusable solutions: promoting the process of synthesizing planning systems. The aim of the Apsi-Trf was to promote and enable this advancement and indeed it allowed the fast realization of several prototypes. During the project for example, three separate groups of scientists have rather quickly developed specific innovative applications: (1) first the above described problem of the Long Term Planning in Mars-Express has been addressed by our team producing MrSPOCK; (2) second the problem of science planning in the INTEGRAL mission has been addressed and a solution proposed by other colleagues [32] has been delivered to ESA that provides a very effective metaheuristic for such a problem built on top of the Apsi-Trf services; (3) the third case focused on the support to Long Term Planning of XMM-Newton [24]. These results demonstrated the effectiveness of the software framework in speeding up the application development. Furthermore we are currently working to insert a planner developed in the APSI platform as a component for deliberative decisions in the GOAC project [20].

3.2 Desired features for space applications

After having outlines some of the general and recurring problems in the development of space applications, in this section we describe some specific features that this applications should possess and that have been a guarantee for success in our particular experience. In particular we refer to features for (a) elaborating with respect to the definition of quality of solutions, (b) providing incremental support to the problem definition and solution synthesis and (c) integrating services for interaction that have proven to be particularly useful for inserting prototypes within real contexts of use.

Solution Quality. The quality of a solution is one of the most important aspect of a planning support system within the space missions environment. The work described in [12], identifies several parameters that contribute to define the quality of a plan. In the case of MVP (Multimission VICAR Planner - MVP) for example, which is an application dedicated to image-processing, an important aspect is the quality of the image output. In particular, a planner should be able to represent and reason about the effects that various transformations of the images may have on the quality of the image itself. This is something related to a specific aspect of the problem but there are other qualities that can be considered as more general.

Efficiency and *flexibility* are, for instance, two measures that also have a large impact on the success of the mission. In particular, the first dimension can positively affect the space agencies' efficiency by reducing costs thanks to an optimized use of the available resources. As for the second aspect, generating robust solutions can help handling any failures or unexpected events, as well as supporting a possible rescheduling need due to emerging problems. The quality of a plan is also connected to the system's ability to generate solutions that respect both the *soft* and the *hard* constraints of the problem as well as *preferences* of the mission planners.

Our experience is in line with the considerations. In particular, in almost all of our applications, as already mentioned, one thing in common is the need to consider divergent objectives for which it is important to find a good compromise.

For this reason the quality of a solution has several aspects that can be viewed as different components of a multi-objective function to be optimized. In the Mexar2 example three main issues were identified to characterize the quality of a solution:

- the main concern of mission planners is to avoid as much as possible potential overwriting (i.e., data loss). Therefore the volume of data *lost* can be considered as the driving *quality measure* for evaluating solutions.

- A second important aspect that characterizes a dump plan is represented by the plan *size*. Indeed, the time needed to upload the dump commands to the spacecraft (during the subsequent phase called uplink), is proportional to the size of the plan. For this reason a short plan is to be preferred to a longer one.

This again can be seen as particularly related to the specificity of the problem.

- As additional quality measure we considered plan *robustness*, which is related to a solution that preserves a specified amount of space for each packet store in order to safeguard against overwriting.

In the case of Raxem and Raxem2 we have chosen a particular measure for assessing the quality of solutions. In particular we defined the *accuracy* of a solution as an estimate of the degree of adherence of the solution to user's expectations, that is, how much the solution complies with user's desire of having all Mdafs planned with a certain uplink and confirmation scheme combination.

As mentioned above MrSPOCK, defines a multi-objective function whose components are mainly three:

Science: optimizing with respect to this metric entails maximizing the activities of science compared with available opportunities; *Downlink*: by optimizing this component of the objective function the downlink opportunities are maximised; *Uplink distribution*: an important aspect of the mission planner 'on even the possibility of having a uniform distribution of opportunities to uplink, i.e., have the opportunity to communicate with the satellite which are numerous and evenly distributed.

Of course, these goals are conflicting and influence one another. For example, both the activities of science, and some maintenance tasks that must be done regularly contrasts the goal of having numerous and uniformly distributed opportunities for the uplink.

MrSPOCK provides the user the possibility of tuning the algorithm parameters so as to favor one or the other component of the objective function. In this way he/she can generate different solutions and select from those available in a database, depending on the need of the moment.

Overall this is a characteristic that we have tried to implement on all of our applications. In fact, in almost all delivered systems we have allowed the user to generate a set of solutions to the problem that were based on different aspects of solution quality, leaving to the human planner the final choice.

Incremental refinement and interaction support. Most of the problems within the space mission environment entails the needs to support incrementally either the problem definition or the solution creation. Both these aspects are related to the need capture the possible unexpected events or any deviation from the normal courses of actions. For this reason the system should be able to support an incremental adjustment of the problem definition, as well as to dynamically create new solutions able to deal with the changes in environment.

The case of the Mex-Up problem, is for instance particularly related to this aspect, being inherently incremental. To manage this incrementally the interaction environment provides a means to define a new problem and change it incrementally, allowing flexibility in the problem specification as well as in its modification to absorb contingencies and unexpected events.

We have realized that support incrementality has a strong role in the success of deployed applications. This feature enables users to perform strategic reasoning and what-if scenarios projection. To effectively support incrementality a powerful interaction layer is necessary, well connected with capabilities of the plan representation module.

One last aspect that is worth mentioning is the users attitude toward the continuous use of decision support tools within space contexts. From one hand we found that users want to be relieved by hard work, entrusting the automated algorithm to deal with technical details and difficult calculations. On the other hand, it is also clear that users want solutions that incorporate their experiences, preferences and expectations, and above all they want a complete control and supervision over the final choice. Somehow we ended up realizing examples of intelligent systems augmented work environments where users can have access to a number of collaborative systems that facilitate their regular activities.

4 Research Directions and Conclusions

In this paper we have tried to analyze our work for ESA over ten years trying to underscore rationale for success factors and get some hints for future developments. We consider a significant achievement having fielded three different end-to-end interactive problem solvers and having established a collaboration with the ESOC Mars-Express mission environment which is traditionally quite conservative. Key factors for success have been: (a) the choice of the timeline-based representation that turned out as important for facilitating the elicitation of relevant features and their constraints, (b) to enable hybrid approaches to problem solving that has injected flexibility also allowing the customization of problem specific biases never easily captured by general solvers, (c) the choice of interactive solutions that preserve role and authority of the human users, and, a more subtle point, (d) proposing approaches which were non disruptive with respect to previous or current mission practice. The last feature is often difficult to describe and very often requires a lot of additional work to be defended, nevertheless plays a key role in technology acceptance.

In the paper we have also introduced the distinction between product vs. process driven approaches. For sure the APSI project have paved the way in a direction which is not only interesting bot somehow inevitable for the Agency. In fact it may represent a way to cope with the natural conservativeness of both its internal environment and the set of industry usually working for it. Additionally the fact that MrSPOCK, a product based on the Apsi-Trf platform, arrived to be operational creates a success story. Nevertheless it is worth saying that the Apsi-Trf is a first step toward the synthesis of software development environments for planning and scheduling systems. It contributes with a number of innovative ideas in an area in which already the notable examples of Europa [16] and ASPEN [13] exist. For example, it is worth noting, as described in [9], that the current Apsi-Trf is the tool that better allows to integrate constraint-based resource reasoning in a timeline based development environment. Indeed, we should say that the platform in its current shape is a significant steps that needs further investments before achieving industrial strength. For sure we can say that it captures several of the modeling features needed in space domains and it is flexible enough to serve different application domains. Nevertheless to develop effective problem solvers, in particular the hybrid ones that are very often needed in space, the ability and know-how of a research group is still difficult to be customized in an effective software engineering tool of this type. Furthermore, the development of a process driven approach to synthesize interaction modules is again out of the current scope. In this respect it is still in the ESA hands the possibility of supporting the Apsi-Trf development through a number of diversified supporting actions.

Acknowledgments

This work has been sponsored by the European Space Agency under several contracts over the years. The authors would like to thank the Agency for such support. Specific people need to be thanked for their contributions, their continuous assistance and, definitely, for trusting us. In particular we would like to thank Nicola Policella, Erhard Rabenau, Jonathan Schulster, Michel Denis and Alessandro Donati. The work concerning APSI has been supported under ESA ITT 5965 2006. APSI partners have been VEGA GmbH, Onera, University of Milan and ISTC-CNR.

References

[1] J. Amilhastre, H. Fargier, and P. Marquis. Consistency Restoration and Explanations in Dynamic CSPs - Application to Configuration. *Artificial Intelligence*, 135(2002):199–234, 2002.

[2] D. Anderson, E. Anderson, N. Lesh, J. Marks, B. Mirtich, D. Ratajczak, and K. Ryall. Human-Guided Simple Search. In *AAAI-00. Proceedings of the 17th National Conference on Artificial Intelligence*, pages 209–216. AAAI Press / The MIT Press, 2000.

[3] G. Bernardi, A. Cesta, and G. Cortellessa. Deploying RAXEM2: Planning Improvements in Daily Work Practice. In *SPARK-09. Scheduling and Planning Applications woRKshop at ICAPS*, Thessaloniki, Greece, 2009.

[4] A. Cesta, G. Cortellessa, M. Denis, A. Donati, S. Fratini, A. Oddi, N. Policella, E. Rabenau, and J. Schulster. MEXAR2: AI Solves Mission Planner Problems. *IEEE Intelligent Systems*, 22(4), 2007.

[5] A. Cesta, G. Cortellessa, M. Denis, A. Donati, S. Fratini, A. Oddi, N. Policella, E. Rabenau, and J. Schulster. Continuous Plan Management Support for Space Missions: the RAXEM Case. In *PAIS/ECAI-08. Prestigious Applications on Intelligent Systems. Proceedings of the 18th European Conference on Artificial Intelligence*, pages 703–707, 2008.

[6] A. Cesta, G. Cortellessa, S. Fratini, and A. Oddi. Looking for MrSPOCK: Issues in Deploying a Space Application. In *SPARK-08. Scheduling and Planning Applications woRKshop at ICAPS*, Sidney, Australia, 2008.

[7] A. Cesta, G. Cortellessa, S. Fratini, A. Oddi, and Policella. N. An Innovative Product for Space Mission Planning – an a posteriori evaluation. In *ICAPS-07. Proceedings of the 17th International Conference on Automated Planning and Scheduling*, 2007.

[8] A. Cesta, G. Cortellessa, A. Oddi, and N. Policella. Studying Decision Support for MARS EXPRESS Planning Tasks: A Report from the MEXAR Experience. In *IWPSS-04. Proceedings of the 4th International Workshop on Planning and Scheduling for Space*, 2004.

[9] A. Cesta and S. Fratini. The Timeline Representation Framework as a Planning and Scheduling Software Development Environment. In 27^{th} *Workshop of the UK Planning and Scheduling Special Interest Group*, pages 17–24, Edinburgh, UK, 2008.

[10] A. Cesta and A. Oddi. Gaining Efficiency and Flexibility in the Simple Temporal Problem. In *TIME*, 1996.

[11] A. Cesta, A.and Finzi, S. Fratini, A. Orlandini, and E. Tronci. Flexible Timeline-Based Plan Verification. In *KI*, pages 49–56, 2009.

[12] S. Chien, T. Estlin, F. Fisher, and G. Rabideau. Lessons Learned Deploying Space Planning Systems. *Applied Artificial Intelligence*, 15(2):161–187, 2001.

[13] S. Chien, G. Rabideau, R. Knight, R. Sherwood, B. Engelhardt, D. Mutz, T. Estlin, B. Smith, F. Fisher, T. Barrett, G. Stebbins, and D. Tran. ASPEN - Automated Planning and Scheduling for Space Mission Operations. In *Proceedings of SpaceOps 2000*, 2000.

[14] P.R. Conrad, J.A. Shah, and B.C. Williams. Flexible Execution of Plans with Choice. In *ICAPS*, 2009.

[15] G. Dooms, P. Van Hentenryck, and L. Michel. Model-driven Visualizations of Constraint-based Local Search. *Constraints*, 14(3):294–324, 2009.

[16] Europa. Europa Software Distribution Web Site. `https://babelfish.arc.nasa.gov/trac/europa/`, 2008.

[17] R.E. Fikes and N.J. Nilsson. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence*, 2(3-4):189–208, 1971.

[18] M. Fox and D. Long. PDDL 2.1: An Extension to PDDL for Expressing Temporal Planning Domains. *Journal of Artificial Intelligence Research*, 20:61–124, 2003.

[19] J. Frank and A. Jonsson. Constraint Based Attribute and Interval Planning. *Constraints*, 8(4):339–364, 2003.

[20] GOAC. Goal Oriented Autonomous Controller. ESA ITT 5961, 2009.

[21] N. Jussien and V. Barichard. The PaLM System: Explanation-based Constraint Programming. In *Proceedings of TRICS: Techniques foR Implementing Constraint programming Systems, a post-conference workshop of CP 2000*, pages 118–133, Singapore, 2000.

[22] N. Jussien and O. Lhomme. Local Search with Constraint Propagation and Conflict-based Heuristics. *Artificial Intelligence*, 139(1):21–45, July 2002.

[23] K.W. Klau, N. Lesh, M. Marks, and M. Mitzenmacher. Human-guided Search. *Journal of Heuristics*, 16(3):289–310, 2010.

[24] M. Lavagna and F. Castellini. Advanced Planning and Scheduling Initiative's XMAS tool: AI for automatic scheduling of XMM-Newton long term plan. In *IWPSS-09. Proceedings of the* 6^{th} *International Workshop on Planning and Scheduling for Space*, Pasadena, CA., 2009.

[25] P.H. Morris and N. Muscettola. Temporal Dynamic Controllability Revisited. In *AAAI*, pages 1193–1198, 2005.

[26] P.H. Morris, N. Muscettola, and T. Vidal. Dynamic Control Of Plans With Temporal Uncertainty. In *IJCAI*, pages 494–502, 2001.

[27] N. Muscettola. HSTS: Integrating Planning and Scheduling. In Zweben, M. and Fox, M.S., editor, *Intelligent Scheduling*. Morgan Kauffmann, 1994.

[28] A. Oddi and N. Policella. Improving Robustness of Spacecraft Downlink Schedules. *IEEE Transactions on Systems, Man, and Cybernetics – Part C Applications and Reviews*, 37(5), 2007.

[29] A. Oddi, N. Policella, A. Cesta, and G. Cortellessa. Constraint-Based Random Search for Solving Spacecraft Downlink Scheduling Problems. In G. Kendall, E. Burke, S.Petrovic, and M. Gendreau, editors, *Multidisciplinary Scheduling: Theory and Applications*, pages 133–162. Springer, 2005.

[30] L. Planken, M. de Weerdt, and R. van der Krogt. P3C: A New Algorithm for the Simple Temporal Problem. In *ICAPS*, pages 256–263, 2008.

[31] N. Policella, A. Cesta, A. Oddi, and S. F. Smith. Solve-and-Robustify. Synthesizing Partial Order Schedule by Chaining. *J. Scheduling*, 12(3):299–314, 2009.

[32] C. Pralet and G. Verfaillie. AIMS: A Tool for Long-term Planning of the ESA INTEGRAL Mission. In *IWPSS-09. Proceedings of the* 6^{th} *International Workshop on Planning and Scheduling for Space*, Pasadena, CA., 2009.

[33] R. Rasconi, A. Cesta, and N. Policella. Validating Scheduling Approaches Against Executional Uncertainty. *Journal of Intelligent Manufacturing*, 21(1):49–64, 2010.

[34] G. Righini and E. Tresoldi. A Mathematical Programming Solution to the Mars Express Memory Dumping Problem. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 40(3):268–277, 2010.

[35] S.F. Smith, G. Cortellessa, D.W. Hildum, and C.M. Ohler. Using a scheduling domain ontology to compute user-oriented explanations. In L. Castillo, D. Borrajo, M.A. Salido, and A. Oddi, editors, *Planning, Scheduling, and Constraint Satisfaction: From Theory to Practice*. IOS Press, 2004.