ECAI 2008
M. Ghallab et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-891-5-703

Continuous Plan Management Support for Space Missions: the RAXEM Case

Amedeo Cesta[a], Gabriella Cortellessa[a], Michel Denis[b], Alessandro Donati[b], Simone Fratini[a], Angelo Oddi[a], Nicola Policella[b], Erhard Rabenau[b] and Jonathan Schulster[b]

[a] **ISTC-CNR, National Research Council of Italy, Institute for Cognitive Science and Technology, Rome, Italy**
⟨name⟩.⟨surname⟩@istc.cnr.it

[b] **ESA/ESOC, European Space Agency, European Space Operations Centre, Darmstadt, Germany**
⟨name⟩.⟨surname⟩@esa.int

Abstract. This paper describes RAXEM, an AI-based system developed to support human mission planners in the daily task to plan uplink commands for an interplanetary spacecraft. The intelligent environment of RAXEM has been designed to support the users in analyzing the problem and taking planning decisions as a result of an interactive process. The system combines different ingredients like integrating flexible automated algorithms, promoting user active participation during problem solving, and guaranteeing continuity of work practice. The paper touches upon all these aspects and comments on how a key factor for success has been the integration of intelligent technology to continuously support mission plan management.

1 Introduction

The space domain is one of those in which AI Planning and Scheduling (P&S) technology has demonstrated maturity and effectiveness. A significant effort has been directed to have advanced examples of autonomy, see Remote Agent [4] and EO-1 [3]. Other very interesting success stories have addressed daily problems at ground segments to support either payload scientists negotiation, like in MAPGEN [1], or mission planners activities, like in MEXAR2 [2].

This paper describes a new system developed to support the command uplink problem (formalized as MEX-UP) for the MARS EXPRESS, a spacecraft which is currently operational in Mars orbit. To solve the MEX-UP problem a planning tool called RAXEM has been developed to support daily mission planners activities. The RAXEM uplink planning tool has been designed and engineered to optimize the safety and timeliness of the more than fifty command timelines sent to MARS EXPRESS each week. The RAXEM tool is in operational use since late Summer 2007. It uses 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. A key point in RAXEM 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 a supporting front-end should be able to allow smooth accommodation of these late requests. To this aim RAXEM has been endowed with an interaction layer that supports incremental plan definition and management. Particularly useful are functionalities to continuously check the situation of the commands already on board to be executed and those that still need to be uplinked.

In this paper we describe both the technological/AI aspects of RAXEM development and the users perspective on the tool. We underscore how the end-to-end features of these systems are contributing not only to support mission operations but to increasingly inject innovative ideas about more flexible ways of managing operations during mission. Projects like RAXEM and those mentioned above, have the merit of increasing the awareness in the operational space environment of the reliability and maturity of AI technology and intelligent systems in general.

2 The MEX-UP Problem

The MARS EXPRESS [1] 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. In particular a particular on-board memory block, called Mission TimeLine (MTL), is replenished by uploading time-tagged telecommands (TCs) from the ground.

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 for each operation a set of time-tagged telecommands are synthesized and then are collected in a set of MTL *Detailed Agenda Files* (MDAF) by the Mission Planning Engineers: this step ensures that all the TCs related to a particular operation/procedure are in the same MDAF and then are all uploaded together. This is fundamental to avoid having the spacecraft in an inconsistent status in case of a transmission failure - we do not want the spacecraft pointing somewhere in the space without knowing what to do next.

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. This situation is shown in Fig. 1: a set of requests has to be sent to the MARS EXPRESS probe through a limited transmission channel in order to define the operations that has to be accomplished. 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. This is the problem named MEX-UP. The goal of our work has been to synthesize a consistent

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

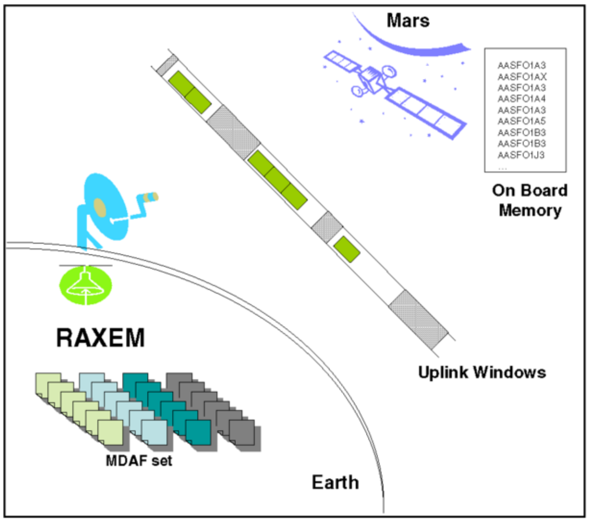

Figure 1. A sketchy representation of MEX-UP

sequence of activities for uploading the set of commands on-board (i.e., the uploading plan).

For a complete formulation of the MEX-UP problem, the following additional aspects need to be taken into account:

– each MDAF is a triple $\langle first, last, size \rangle$ that defines respectively, the execution time of the first telecommands, the execution time of the last telecommands, and the number of telecommands in the MDAF;
– each MTL is a pair $\langle size, cache \rangle$ that defines respectively, its size and the size of its cache (i.e., the most immediate TCs to execute);
– each uplink window is a n-ple $\langle st, et, dur, owlt \rangle$ that defines respectively, the start time, the end time, the duration, and the one-way-light-time, OWLT;[2]
– finally, given an MDAF m, the duration of the uplink process is a function $dur_{up}(m)$ that depends on the number of TCs in m, the execution time of the first TC, the uplink window available, and the MTL status.

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.

Technical/Managerial Constraints. In addition to the basic features of the problem listed above, there are a number of additional aspects that suggested the users to look for a more intelligent tool able to explore more deeply the solution space. These issues concern both additional properties required for the solution and some existing problems in work practice. From the point of view of the distinguishing qualities of a solution we have:

– the possibility of choosing, for each MDAF, among different uplink modalities with an impact on the duration of the associated uplink activity. In particular three modalities can be considered: (a) *full confirmation*, in this case the activity requires the time needed for transferring on board the file, performing a specific processing procedure, and receiving the acknowledgement; (b) *reduced confirmation*, in this case the constraint of waiting for the processing procedure is relaxed. Then the activity requires just the time needed for transferring on board the file and receiving back the acknowledgement; (c) *no confirmation*, this extreme modality consists in relaxing also the need for the acknowledgement. The activity duration is then equal to the time needed for transferring the MDAF.
– Additionally, the users may merge a group of heterogeneous MDAFs in order to optimize the use of the transmission windows (the merging allows saving of the OWLT). This is a further flexibility service for the users to explore types of solutions.
– The need for identifying alternative uplink windows for each MDAF. This is necessary to support both a possible uplink failure and to manage the case that the reserved ground station is not available. The last case can happen when the ground station is reallocated at late notice to another mission with higher priority then MARS EXPRESS.

Other constraints concerns more explicitly the modality of work. The reasons mission planning engineers were looking for a supporting tool stem in the fact that (a) planning uplink is a continuous, time-consuming task that, even though very important for the mission, tends to become routine hence inevitably prone to errors; (b) uplink is incremental so there is a need to manage dynamically a current plan and very easily the type of solutions may become "patched", thus entailing a decreased quality in the solution. On the contrary a problem solver can explore the space of solutions more effectively; (c) there is always need to insert additional command sets to upload for some emergency or unforeseen events. Also in this case the possibility of computing an automated solution quickly and maintaining quality is important. Finally, (d) the introduction of a plan management tool allows flexibility not only in exploring the solution space but also to support feature changes so as to reflect the incrementality of the problem.

To sum up, MEX-UP is a planning problem that should respect a lot of specific mission constraints that impact the quality of alternative solutions. The satisfaction of the users is grounded on the possibility of exploring alternative solutions. Additionally a clear example is given of how the problem is a combination of problem solving and plan management. Satisfaction of the user is connected to the integration of different services in the plan life cycle demonstrating how planning services should involve more than the single solving functionalities [6].

3 The RAXEM Tool

RAXEM has been developed (and then introduced in the work practice) when the mission was already operational since a long period; this has required preserving the mission planning work practice as much as possible. The goal was to address some weaknesses of the working cycle that were considered minor at mission design time and ended up having an impact on the quality of work during daily mission operations. We have developed an end-to-end application with particular attention to the maintenance of the mission data flow and to the idea of preserving key decisions for mission planners.

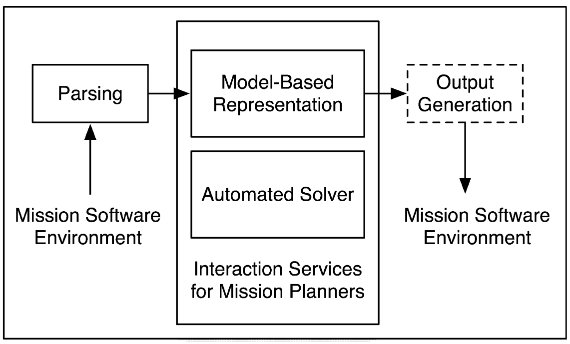

Figure 2. The general software architecture

[2] The OWLT is the elapsed time it takes for light (or radio signals) to travel between the Earth and a celestial object (in this case Mars).

RAXEM directly accepts as input the MDAF files and the Uplink Window Files (UWFs — describing the temporal availability of different ground stations), and produces uplink files in the expected format for the MARS EXPRESS data cycle. This is obtained by encapsulating the intelligent system between two software modules (see Fig. 2): (1) the *Parsing* module that processes the input files and selects the relevant information for the symbolic model used by the solver, and (2) the *Output Generation* module that manipulates the results produced by the system and generates the output according to external formats. Fig. 2 shows a complete blow up of RAXEM software components. Apart the two input/output modules[3] the system involves three modules that provide the core functionalities: (a) a domain modeler (Model-Based Representation in the figure), (b) an algorithmic module (Automated Solver), (c) an interaction module (Interaction Services) that allows mission planners to access the other two modules. The Parsing and Output Generation modules directly interact with the model-based representation that acts as the key module for the whole approach. The solver directly extracts and stores all the relevant information from/to the modeling part.

3.1 Modeling with Timelines

As in any AI approach the basic step in solving the MEX-UP problem has been to build a representation (or *model*) of the domain which contains the relevant objects and constraints that influence the problem solving in the particular domain. In particular we have followed a timeline-based approach [5, 4, 3] which focuses the attention on problem features evolving over time. Deciding on temporal evolution of the main timelines is the "meta-goal" of the problem solver. The representation choices have been fundamental because they not only support the solving algorithm but are at the base of the interaction with the user.

In RAXEM we consider the temporal evolution of the two relevant system components:

– *Mission Timeline (*MTL*)*. The MTL contains the set of telecommands. This can be represented as a *cumulative* resource characterized by a finite capacity and a finite cache capacity.
– *Communication Channel*. The uplink connection to Earth for transmitting data. This resource, which is *binary* (either busy or free), is characterized by a set of separated transmission windows which identify time intervals for uplink.

In this way we restrict the problem to consider the resource profiles of the MTL and the channel availability for uplink. The core of the MEX-UP problem is to decide the *uploading plan* that is when each MDAF can be uploaded.

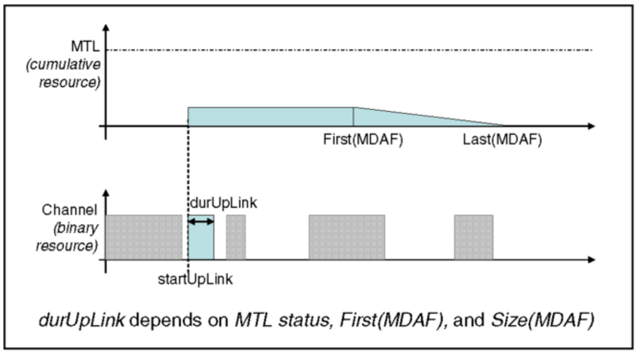

Figure 3. The main problem timelines

[3] In the figure, the Output Generation is represented with a dotted line because the task is performed externally from the current RAXEM system.

As shown in Fig. 3, for each MDAF ready to uplink it is possible to identify two different activities to allocate on the previous component timelines:

– An *Uplink activity*: this is to represent the transmission over a free slot of the communication channel. This operation will require the whole bandwidth of the communication channel for the entire duration (due to the binary capacity). As said in the figure the duration of the transmission depends on the execution time of the first telecommand in the file, the size of the MDAF, and on the MTL "status". Such a status is particularly relevant when the MTL is almost full to capacity, or some last-minute set of commands should be allocated directly in the on-board cache;
– An MTL *operation*: at its start time, each operation "instantaneously" stores in the Mission Timeline an amount of data equal to the number of telecommands in the MDAF, Size(MDAF). At the specified execution time, each TC will be released and removed from the MTL. In the figure a linearly decreasing behavior is given for a certain MDAF, this is not the general shape of the curve because TC execution time depends on the TCs distribution within a given MDAF.

The figure describes the basic synchronization constraints that are immediately translated in temporal constraints to be satisfied. In addition the solver should both cope with the additional Technological constraints described before and comply with a number of local decisions posted by the users.

3.2 Problem Solving Capabilities

To better cope with the detailed constraints in the problem (uplink mode, need to reserve a secondary window, etc.) we have designed a two steps algorithm. Basically the algorithm starts trying to fulfill all the MDAF requirements and in case no solution is available it incrementally relaxes the different requirements in order to obtain a feasible uplink plan.

The *first step* iteratively produces an initial uplink plan. In input we have a set of files of telecommands (MDAFs) to be allocated, a set of MDAFs already allocated (necessary to know the initial situation of the on-board memory), and a set of communication windows. For each MDAF we have to decide (1) which uplink modality to use, (2) if the file can be associated with other MDAFs in a unique uplink activity or it has to be uploaded alone, and, of course, (3) at what time instant to start the file uplink (as described above the activity duration depends on the uplink time, the number of TCs in the file, and the state of the MTL). Two are the fundamental constraints to be taken into account: the availability of a sufficient communication window and the availability of sufficient room in the MTL in order to allocate the telecommands contained in the file. Therefore in the first step, the files are sorted according to the execution time of their first telecommand (*firstTC*). Given this order each MDAF is allocated in order to be uploaded in a Multi-MDAFs activity, with a full confirmation uplink mode, and with a backup (or secondary) window. In case one of the MDAFs cannot be uplinked (either in a Multi-MDAFs or single-MDAF uplink activity) in a full confirmation mode the algorithm relax this constraint in order to complete the solution (first from full confirmation to reduced confirmation and then, in case of failure, from reduced confirmation to no confirmation).

In case the first step does not produce a complete solution (i.e., all the files are allocated for uplink), the *second step* aims at completing the current plan. The algorithm is a complete search that at each step removes a previous decision (file planned to be uplinked) in order to find space for the activities that have not been planned. The goal is to maximize the number of files to uplink and in case of solutions

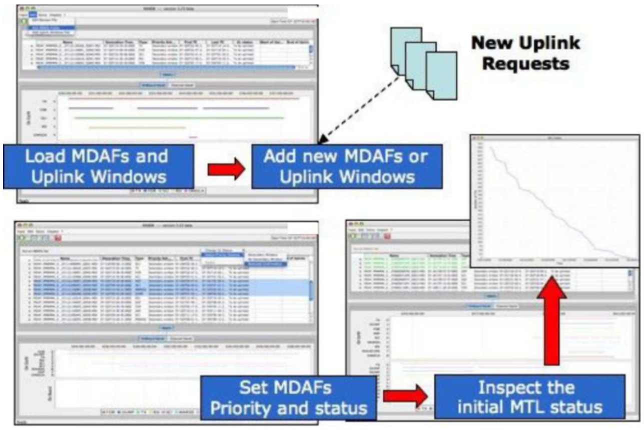

Figure 4. An example of workflow for incremental problem definition

with the same number of uplinked files, to maximize the number of telecommands uplinked.

It is worth noting that a theoretical alternative to this second phase would consist in considering also complete solutions and trying to optimize it. Unfortunately operative constraints require that the *firstTC* order of allocation should be respected (also penalizing the optimality of the solution). This makes this alternative approach not viable in practice.

3.3 Interactive Plan Management

As already mentioned the core functionalities of the RAXEM architecture include a layer for user interaction. This aspect is particularly relevant for addressing the MEX-UP application because of the importance of the requirements for plan management within the whole set of problem features. Plan management is grounded on the *model based representation* whose functionalities are instrumental not only to the *automated solver* but also to guarantee a level of "understandability" to the services toward the users. The interaction services support the mission planner during the whole *uplink-plan life-cycle* providing an environment to support three main tasks: (a) *incremental management of the problem*; (b) *inspection of plans*; (c) *what-if projection*. Furthermore, in designing this environment we paid attention to both reproduce the previous work practice – so as to foster a seamless integration within the working environment – and augment mission planners capabilities with the aim of improving work efficiency and solutions quality.

Incremental problem definition. The MEX-UP problem is inherently incremental. For example, MDAFs become dynamically available to mission planners, uplink windows may vary according to various availability factors. In addition new requests for uplink may arrive during operations or existing ones may be removed or delayed by users. To manage this incrementality 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. The MDAFs table performs a preliminary input checking and enables users to modify MDAFs priority/type and add/remove MDAFs to be uplinked. Fig. 4 shows an example of task flow during a phase of incremental problem definition. The basic interaction layout shows on top the list of MDAFs, on bottom their location subdivided between on-ground (to be planned for uplink) and on-board (successfully uplinked). The use of colors allows an immediate identification of their type. As new requests for uplink arrive they are loaded incrementally and displayed to the user. The user can change, for example, the relative priority of each of the files influencing the problem solving phase. Furthermore the users may inspect the status of the on-board MTL.

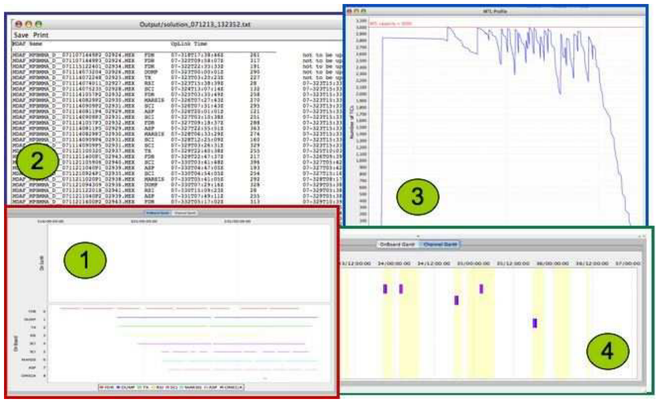

Figure 5. Different views of a solution

Plan inspection. Alternatives views and aspects of the solution are presented to the user for inspection as it is shown in Fig. 5. The box (1) of the picture shows the MDAFs product for a problem whose solution transfer all the MDAFs on board, box (2) shows the solution uplink-plan which associates start and end time for uplink for each MDAF, box (3) contains the MTL status after uplink for the current problem. Alternative information is provided in box (4) which shows the uplink activities subdivided by ground stations and gives also an immediate view of the amount of use of the visibility windows. The visual environment represents a powerful way to check the validity the solutions and allows discovering duplicated files or missing uplink products. If the user is not satisfied with the solution, he/she can change input setting and run the RAXEM tool to obtain different uplink plans that take into account different priorities or new uplink needs.

A specific additional service allows the user to ask for a snapshot of the status of the memory at a given time. The related window, foreground of Fig. 6, displays the list of single telecommands start times and their associated MDAFs. Fig. 6 shows also another possible way for using the inspection modalities: from observing a specific MDAF on the MTL status the user can go for inspection to the MTL resource profile then to the uplink activity on the solution Gantt.

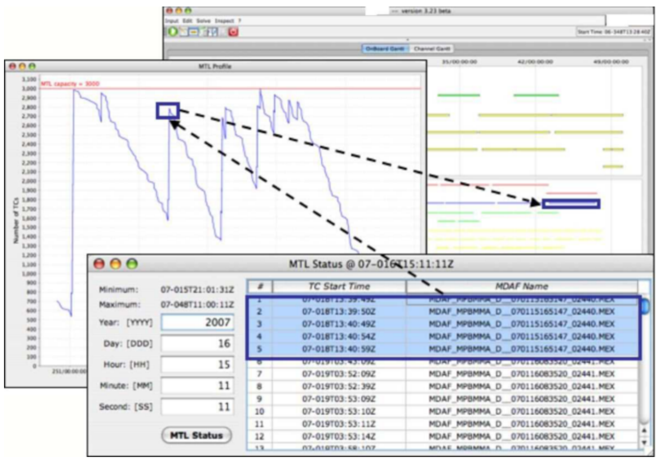

Figure 6. Inspecting on-board status

What-if projection. In addition to the classical use of RAXEM the interaction layer is also instrumental for some form of *what-if* analysis support. Indeed the visual representation of the uplink plans can be used to detect uplink windows used to capacity or MDAFs that have been downgraded by the automated solver from "uplink with secondary window" to "primary window only" . This allows predicting "bottlenecks" in the uplink capability for the mission. With forecasting at medium-term planning level this make it possible to release excess station time not required for uplink of products (or downlink of science data) with consequent cost-savings. Also, the user can evaluate the effectiveness of alleviating particular bottlenecks in the uplink plan by choosing which MDAF to downgrade.

4 An Evaluation from Users

Overall the RAXEM tool has shown very positive outcomes with respect to performance, reliability and actual benefits with regard to planning of the uplink stream for MARS EXPRESS. Since becoming operational in the late Summer 2007 the tool has generated error-free plans for the uplink of all products. The reduction in work-effort for planning one weeks uplink is estimated as about 4-6 hours per week saving. The actual plan is significantly more robust including accurate uplink window timings and a secured alternative uplink window for each product on a separate ground station. The tool also has benefits in terms of configuration control and traceability of uplinked commanding files, as well as it allows almost effortless re-planning in the case of single MDAF modification and/or addition of new MDAFs after the normal planning cycle is complete.

The main achievements of the RAXEM tool can be summarized as follows:

Safety and security. RAXEM has achieved its stated objectives of maintaining the on-board command queue (Mission Timeline or MTL) as full as possible, while ensuring safety of the commanding chain through provision of fully redundant uplink opportunities on two different ground stations for each product wherever feasible. This in principle provides improved security and safety for mission integrity even in the event of the total outage of one ground station. The work-hours involved in planning the uplink for a week has been reduced by a factor of 4-6 on average, depending on the complexity of the planning task – the more uplink products and the shorter or more infrequent the uplink-windows the greater the saving, since RAXEM takes the same time to run regardless of problem complexity. The checking of the uplink solution still takes longer with a more restrictive uplink case. Also on the operations standpoint it allows a fast response time to restart science operations after a Safe Mode of the spacecraft,[4] whereby all MDAFs must be re-sent to populate an empty MTL.

Efficiency and accuracy. Another benefit is that re-planning an uplink solution if an additional file is added or one is replaced or deleted takes very little time. Most of the effort is in checking the solution that RAXEM proposes but usually only the "deltas" need to be rechecked. The RAXEM tool has greatly improved the quality and accuracy of the uplink requests, by eliminating the human errors that occasionally occurred in completing forms by hand for files with long and similar names. The tool is very easy to use, and training of a new user from scratch can be completed usually within an hour, with very little follow up support required for the typical experienced engineer. RAXEM provides a powerful visualization interface that allows rapid checking for duplicated files or missing uplink products which show up as 'gaps' in the timeline for a particular product stream.

Flexibility and traceability. The tool allows forward modelling and prediction of "bottlenecks" in the uplink capability for the mission – the graphical representation clearly shows where all available uplink windows are being used to capacity or products have been downgraded from "uplink with secondary window" to "primary window only". With forecasting at medium-term planning (monthly) level it will be possible to release excess station time not required for uplink of products (or downlink of science data) with consequent cost-savings on ground station allocations for which charges per hour are made against the mission budget. RAXEM also ensures full traceability of all uplinked products, right from generation of the commanding to actual execution on board the spacecraft. Since introducing RAXEM we have had no missed uplink of products due to "human error", where a file was missed out or uplinked twice. This significantly improves the overall safety and security of the mission operations.

5 Discussion and Conclusions

The RAXEM experience reinforces some remarks coming from other work on innovation infusion within space environments. It showed the capability of AI technology to increase in performance the management of specific aspects of the mission planning process, e.g., save time and money, reduce human error, increase science data return, improve the solution robustness, etc. Additionally we have experienced the importance of a global approach to the problem, which entails not only the production of a smart algorithm but also the design of a complete tool to support users in charge of the problem. This effort in synthesizing a "complete application" can be identified as the key feature when proposing solutions to mission planners who are supposed to work on a problem day by day for the entire period of a mission. This aspect is particularly relevant in the case of RAXEM where managing continuity in operation, incrementality in problem definition, reaction to problem changes and monitoring of current status are basic needs of the environment.

The introduction of RAXEM within the operational contexts has made clear how very often single problems could be tackled in a more integrated way and with a more systematic approach. In current practice RAXEM is inserted within a work cycle together with other specific tools that manipulate input and output of RAXEM and contribute to a comprehensive "integrated uplink service support line".

In closing the paper we quote a recent informal comment from the users: "despite some initial skepticism that such an AI-based tool would have been able to improve on the performance of a team of highly-experienced engineers, acceptance of RAXEM was so widespread that since its first introduction (Summer 2007) no one has made use anymore of the option to plan the uplink by hand!".

REFERENCES

[1] M. Ai-Chang, J. Bresina, L. Charest, A. Chase, J. C. Hsu, A. Jonsson, B. Kanefsky, P. Morris, K. Rajan, J. Yglesias, B. G. Chafin, W. C. Dias, and P. F. Maldague, 'MAPGEN: Mixed-Initiative Planning and Scheduling for the Mars Exploration Rover Mission', *IEEE Intelligent Systems*, **19**(1), 8–12, (2004).

[2] 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), 12–19, (2007).

[3] S. Chien, R. Sherwood, D. Tran, B. Cichy, G. Rabideau, R. Castano, A. Davies, D. Mandl, S. Frye, B. Trout, S. Shulman, and D. Boyer, 'Using autonomy flight software to improve science return on earth observing one', *Journal of Aerospace Computing, Information, and Communication*, **2**(4), 196–216, (2005).

[4] A.K. Jonsson, P.H. Morris, N. Muscettola, K. Rajan, and B. Smith, 'Planning in Interplanetary Space: Theory and Practice', in *AIPS-00. Proceedings of the Fifth Int. Conf. on Artificial Intelligence Planning and Scheduling*, (2000).

[5] N. Muscettola, S.F. Smith, A. Cesta, and D. D'Aloisi, 'Coordinating Space Telescope Operations in an Integrated Planning and Scheduling Architecture', *IEEE Control Systems*, **12**(1), 28–37, (1992).

[6] M. E. Pollack and J. F. Horty, 'There's More to Life than Making Plans: Plan Management in Dynamic, Multi-Agent Environments', *AI Magazine*, **20**(4), 71–84, (1999).

[4] This is a precautionary condition in which all the science operations are stopped in order to cope with particular events (e.g., eclipse seasons).