Studying Decision Support for MARS EXPRESS Planning Tasks: A Report from the MEXAR Experience *

Amedeo Cesta, Gabriella Cortellessa, Angelo Oddi and Nicola Policella

ISTC-CNR, Italian National Research Council
Viale Marx 15, I-00137 Rome, Italy
{ a.cesta | corte | a.oddi | policella }@istc.cnr.it

Abstract. This paper describes an experiment of technology infusion into a European Space Agency (ESA) mission. It reports a study conducted in the context of the MARS EXPRESS mission to address the Memory Dumping Problem of the spacecraft. The paper describes the steps for developing a complete approach aimed at creating an interactive decision aid for the human mission planner called MEXAR. In particular, it is shown how problem solving technology for planning, scheduling and constraint reasoning is integrated with an interaction module to create a set of advanced services for the user.

1 Introduction

Between November 2000 and July 2002 the Planning & Scheduling Team at ISTC-CNR has conducted a study for ESA called "Efficient Planning Algorithms for an Interplanetary Mission" (`http://mexar.istc.cnr.it/`). The result of this work was an advanced demonstrator which provides a decision support system for the operations center mission planners. The study aimed at demonstrating the ability of planning and scheduling (P&S) technology to cope with a real space problem in the context of MARS EXPRESS (ESA 2002) mission planning. We started investigating the mission *life-cycle* to identify a subproblem that actually was in need of automation and focused on the traditional management of the data return problem, an example of a mostly "hand made" activity where the human mission planner works at a very low level of abstraction. In particular, for the entire duration of the mission, a human mission planner should synthesize the memory dumping spacecraft commands, an activity which occupies about 40% of his or her time on a continuous basis. We pursued the idea of designing and implementing a decision aid prototype to support the solution of this subproblem. The interactive nature of the tool was intended to show that P&S technology can contribute to facilitate the critical activity of mission planners while preserving their responsibility in taking critical decisions.

Figure 1: One of the first images taken by MARS EXPRESS of the Martian surface — Courtesy of ESA

In May 2002, the final version of the software system, named MEXAR, has been delivered to ESA-ESOC. It is capable of automating the generation of spacecraft operations to downlink the on-board mass memory. The software shows an example of practical integration of problem solving techniques known as CSP (Constraint Satisfaction Problem solving) with interactive techniques from HCI (Human Computer Interaction). This paper is a final report which aims at showing how the approach used to achieve a complete prototype is general enough to be reused in different missions.

The paper is organized as follows: a description of the problem is given, then an architecture for interactive problem solving is sketched and subdivided into two main modules (the problem solver and the interactor). Then details on each of the modules are given, the main design choices

* This work describes results obtained in the framework of a research study conducted for the European Space Agency (ESA-ESOC) under contract No.14709/00/D/IM. Opinions expressed in this paper are those of the authors and do not necessarily reflect opinions of the funding agency.

described and specific features highlighted. A discussion section gives comments on the work, and some conclusions end the paper.

2 Problem Summary

MARS EXPRESS is space probe launched by the European Space Agency (ESA) on June 2, 2003 that has been orbiting around Mars since the beginning of 2004 for two years. It contains seven different scientific payloads that are gathering different data on both surface and atmosphere of the Red Planet (see for example the image in Figure 1). A team of people, the Mission Planners, are responsible for the on board operations of MARS EXPRESS. They receive input from different teams of scientists, one from each payload, and cooperate with different specialists for more specific tasks (e.g., Flight Dynamics (FD) experts). Any single operation of a payload, named POR from Payload Operation Request, is decided well in advance through a negotiation phase among the different actors involved in the process (e.g., scientists, mission planners, FD experts). The result of this negotiation is (a) acceptance or rejection of a single POR, (b) a start time assignment of the accepted PORs. At the operational level the mission planners are responsible for data return to scientists for any single POR. Their goal is to guarantee an acceptable turnover time from the end of the execution of the POR to the availability on Earth of data generated by that POR.

During a certain time window (e.g., two days of operations), the spacecraft produces a large amount of data which derives both from the activities of its payloads —that execute PORs— and from on-board device monitoring and verification tasks —the so called *housekeeping data*. All these data, usually referred to as *telemetry*, are to be transferred to Earth during downlink connections. MARS EXPRESS is endowed with a single pointing system, thus during regular operations, it either points to Mars and performs payload operations or points to Earth and transmits data through the downlink channel. As a consequence on-board data are first stored on the Solid State Mass Memory (SSMM) then transferred to Earth during *temporal visibility windows*. An effective management of on-board memory and a good policy for downlinking its data are very important for a successful operation of the spacecraft. Goal of MEXAR is to offer support for synthesizing downlink operation to dump the SSMM on a regular basis. We have formalized this problem as the MARS EXPRESS Memory Dumping Problem (MEX-MDP) and shown its computational complexity in (Cesta *et al.* 2002; Oddi *et al.* 2003). This paper is dedicated to show how a comprehensive approach for decision support has been built and integrated in a tool called MEXAR that solves MEX-MDP instances.

3 Toward an Interactive Architecture

Given the goal of developing a decision aid for supporting the human mission planner in solving MEX-MDP problems we have chosen to design a software architecture that captures the problem life-cycle. In particular the tool supports its user in all the steps which go from the definition of a MEX-MDP instance to the generation of solutions and refinement. In the current work practice at ESA the user (mission planner) and the spacecraft (MARS EXPRESS) interact through certain modalities. One of the goals of our study has been to contribute an additional means to this interaction offering a tool that preserves completely the “traditional” real world practice and potentially provides new aids.

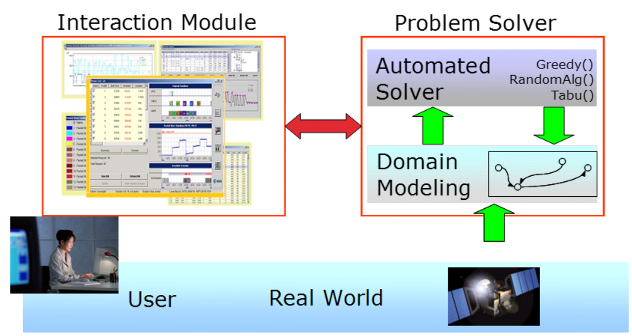


Figure 2: The Interactive Problem Solving Architecture

A software architecture for interactive problem solving that copes with this problem is sketched in Figure 2. The figure shows our general approach that consists in adding a path from the user to the controlled spacecraft. This enhanced path is created through a bipartite architecture composed of a Problem Solver and an Interaction Module. The two modules have distinct roles:

– *Problem Solver*. This module allows to model a MEX-MDP problem and computes one or more solutions.

– *Interaction Module*. This second component is responsible for the dialogue, either simple or sophisticated, with the user. It allows the user to understand what the solver is doing, improving her/his trust in the automated solving activity and providing various levels of intervention in the solving process.

At this level of generality the software architecture can describe any software process. The key issue here is to understand how the two modules have been integrated and what is “inside the boxes”. A key aspect of the authors’ approach is to build a representation (or *model*) of the domain that contains the relevant objects to describe the MEX-MDP problem features. This representation phase is fundamental because a good modeling choice not only supports the solving algorithm but creates a basis for the interaction with the user (so it is the core of the architecture in Figure 2). In MEXAR the domain modeling is grounded on a CSP (Constraint Satisfaction Problem (Tsang 1993)) approach. It is worth reminding that the approach based on a symbolic domain model is widely used in knowledge intensive approaches to P&S like in ASPEN (Chien *et al.* 2000), RAX-PS (Jonsson *et al.* 2000), COMIREM (Smith *et al.* 2003), O-OSCAR (Cesta *et al.* 2001) even though each of these systems gives emphasis to different aspects.

The domain modeling choice influences the family of algorithms that work on the model to solve a specific problem instance. The separation of the model from the algorithm (see Figure 2) in this particular case is also functional to the development of the interface with the user. According to this methodology the user is shown a representation of the relevant components of the domain model, and can explore how the temporal evolution of these components can bring about a solution to a problem, independently from how this solution has been achieved.

To sum up in our work we pursue the idea that a user is part of the real world and MEXAR endows her with an additional "lens" to analyze the world. We believe that the framework ⟨Problem Solver, Interaction Module⟩ can be considered a quite general schema to follow in the development of decision support systems. It is worth stressing that the two subcomponents require a comparable effort to develop useful and efficient support systems for real application problems.

4 Solving the Memory Dumping Problem

This section gives a view of the effort needed to develop the problem solving module. As said before, this module has the basic task of creating an "internal model" of the real domain, such a model allows to represent different problem instances that will be manipulated by the solution algorithms that produce solutions or declare failures.

Ontology for Domain Modeling. In developing a model of the real world we need to single out the relevant objects used in the representation. To describe a MEX-MDP domain we have used a basic ontology with two classes of objects: *resources* and *activities*. Resources represent subsystems able to give services, and activities model tasks to be executed on such resources. In addition, a set of *constraints* refines the relationships between the two types of objects. Three types of resources are modeled:

– *Packet Stores.* The on-board memory is subdivided into a set of separated packet stores pk_i which cannot exchange data among each other. Each one has a fixed capacity c_i and can be assigned a priority value to model different relevance of their data content. Each packet store, which can be seen as a fixed size file, is managed cyclically: when it is full the older data are overwritten. Within a packet store, data is segmented into data packets.

– *On-Board Payloads.* An on-board payload can be considered as a *finite state machine* in which each state has a different behavior in generating observation data (i.e., in each possible state the payload has a different generation data rate).

– *Communication Channels.* These resources are characterized by a set of separated communication windows identifying intervals of time for downlink. Each temporal window has a constant data rate.

Activities describe *how* resources are used. Each activity a_i has an associated execution time interval, which is identified by its start-time $s(a_i)$ and end-time $e(a_i)$. Each activity is characterized by a particular set of resource requirements and constraints. MEX-MDP includes three types of activities:

– *Payload Operations.* A payload operation por_i corresponds to a scientific observation. Each por_i generates a certain amount of data which is decomposed into different *store* operations according to the MARS EXPRESS operational modalities, and distributed over the set of available packet stores.

– *Continuous Data Streams.* The particular case of the continuous data stream operations cds_i is such that $s(cds_i) = 0$ and $e(cds_i) = +\infty$ (where $+\infty$ is internally represented as a finite temporal horizon). This activity represents a continuous generation of data with a fixed average data rate (it is used to model housekeeping). Indeed we represent a cds as a periodic sequence of store operations. In particular, given cds with a flat rate r, we define a period T_{cds}, such that, for each instant of time $t_j = j \cdot T_{cds}$ ($j = 1, 2, \ldots$) an activity st_{ij} stores an amount of data equal to $r \cdot T_{cds}$.

– *Memory Dumps.* A memory dump operation md_i transfers a set of data from a packet store to a transfer device. Those activities represent the transmission of the data through the communication channel.

With this set of objects we create a symbolic model of the external world (usually called *domain model*) which is used to represent different problem instances. Now that we have detailed some basic modeling choices, we can give a more detailed description of the MEX-MDP problem.

Problem Definition. Given a set of memory store operations from both the scientific observations $POR = \{por_1, por_2, \ldots, por_n\}$ and the housekeeping $CDS = \{cds_1, cds_2, \ldots, cds_m\}$ a *solution* to a MEX-MDP problem is a set of *memory dumps* $S = \{md_1, md_2, \ldots, md_s\}$ such that the following constraints are satisfied:

- The whole set of on board data is “available” on ground within a temporal horizon $\mathcal{H} = [0, H]$.
- Each dump operation starts after the generation of the corresponding data. For each packet store, the data is moved through the communication channel according to a FIFO policy.
- Each dump md_i has an assigned time window $w_j = \langle r_j, s_j, e_j \rangle$, such that the dumping rate is r_j and the constraint $s_j \leq s(md_i) \leq e(md_i) \leq e_j$ holds. Dump operations cannot reciprocally overlap.
- For each packet store pk_i and for each instant t within the considered temporal horizon, the amount of stored data has to be below or equal to its capacity c_i (no overwriting is allowed).

The solutions should satisfy a quality measure. According to requirements from ESA personnel a high quality plan delivers all the stored data as soon as possible according to a definite policy or objective function. To build an effective objective function for this problem the key factor is:

- the turnover time of a payload operation por_i: $tt(por_i) = del(por_i) - e(por_i)$, where $del(por_i)$ is the delivery time of por_i and $e(por_i)$ is the end time of the payload operation on board;

An objective function which considers this item is the *mean α-weighted turnover time* MTT_α of a solution S:

$$MTT_\alpha(S) = \frac{1}{n} \sum_{i=1}^{n} \alpha_i \; tt(por_i) \tag{1}$$

Given an instance of a MEX-MDP, an *optimal solution* with respect to a weight α is a solution S which *minimizes* the objective function $MTT_\alpha(S)$. The weight α can be used to take into account two additional factors: the data priority and the generated data volume. This particular objective function has been chosen after interaction with ESA personnel who requested as the optimization factor in MEXAR the average turnover time. More recently we have investigated different optimization factors and also developed different symbolic models for MEX-MDP in (Oddi and Policella 2004). In the ESA study reported in (Cesta *et al.* 2002) we have developed a solution for this problem using a CSP approach.

A CSP Approach. A CSP instance involves a set of variables $X = \{x_1, x_2, \ldots, x_n\}$ in which each element has its own domain D_i, and their possible combinations are defined by a set of constraints $C = \{C_1, C_2, \ldots, C_m\}$ s.t. $C_i \subseteq D_1 \times D_2 \times \cdots \times D_n$. A solution consists in assigning to each variable one of its possible values s.t. all the constraints are satisfied. A CSP representation of a problem should focus on its important features. In the case of MEX-MDP, we selected the following characteristics: (1) the temporal horizon $\mathcal{H} = [0, H]$, (2) the store operations that are characterized by their start time t and their amount of data d, (3) the temporal windows in which no communication may occur, (4) the finite capacity c of each memory bank, (5) the FIFO behavior of the memory banks.

Considering the first three items we split the temporal horizon $\mathcal{H}$ in different contiguous temporal windows according to *significant events*: store operations and change of transmission rate. The idea is to create a new window for each significant event on the timeline. This partitioning allows us to consider a temporal interval w_j in which store operations do not happen (except for its upper bound) and the data rate is constant. Furthermore, the packet stores' behavior allows us to perform an important simplification. In fact, it is possible to consider both the data in input and those in output to/from the memory as flows of data, neglecting the information about which operations that data refer to (such information can be rebuilt with a straightforward post-processing step). Thus, the decision variables are defined according to the set of windows w_j and to the different packet stores. In particular we consider as *decision variables* δ_{ij}, the amount of data dumped from the packet store pk_i within the window w_j. According to the partition in separate windows we introduce also: (a) d_{ij}, the amount of data stored in pk_j at t_i, (b) l_{ij}, the available capacity of pk_j at t_i, (c) b_j, the maximal dumping capacity within the window w_j. All these items represent the input of the problem.

A fundamental constraint captures the fact that for each window w_j the difference between the amount of generated data and the amount of dumped data cannot exceed l_{ij} (*overwriting*). Additionally, the dumped data cannot exceed the generated data (*overdumping*). We define the following inequalities as *conservative constraints*.

$$\begin{array}{l} \sum_{k=0}^{j} d_{ik} - \sum_{k=1}^{j} \delta_{ik} \leq l_{ij} \\ \sum_{k=0}^{j-1} d_{ik} - \sum_{k=1}^{j} \delta_{ik} \geq 0 \end{array} \qquad i = 1 \ldots n, \; j = 0 \ldots m \tag{2}$$

A second class of constraints considers the dumping capacity imposed by the communication channel. The following inequalities, called *downlink constraints*, state that for each window w_j it is not possible to dump more data than the available capacity b_j.

$$0 \leq \sum_{i=1}^{n} \delta_{ij} \leq b_j \quad j = 1 \ldots m \tag{3}$$

This formalization of the problem as a CSP is directly mapped into a data structure called *Constraint Data-Base*.

This data structure directly supports the choices made by the solving algorithm by performing deductions based on propagation rules derived from constraints (2) and (3).

Solving Algorithms. To solve a MEX-MDP represented as described in the previous section we have developed a two-stage approach: (a) a *Data Dump Level* figures out an assignment for the set of decision variables δ_{ij} such that the constraints from MEX-MDP, (see (2) and (3)), are satisfied; (b) a *Packetization Level*, which is a constructive step. It starts from the solution of the first step and synthesizes the single data dumps within each of the windows w_j (that is, each δ_{ij} of the previous phase is translated into a set of dump activities).

MEXAR is endowed with a multi-strategy solver implemented on the basic blackboard represented by the Constraint Data Base. In particular, we have two algorithms, a greedy and a randomized algorithm, which compute new solutions for a MEX-MDP. A third, tabu search algorithm is used to look for local optimizations on a current solution obtained by the first two approaches. The *Greedy Algorithm* simply consists in assigning a value to each decision variable according to a heuristic. The variables are selected considering the windows in increasing temporal order. Two different solving priority rules are implemented: (a) CFF (Closest to Fill First) selects the packet store with the highest percentage of data volume. (b) HPF (Highest Priority First) selects the packet store with the highest priority. In case a subset of packet stores has the same priority, the packet store with the smallest store as outcome data is chosen. The *Randomized Algorithm* implements a basic random search that turns out to be quite effective in this domain. This method iteratively performs a random sampling of the search space until some termination criteria is met. In our approach we select the variable in a random way, then the maximal possible value is assigned, considering: (1) the data contained in the packet store, (2) the amount of data already planned for dumping and (3) the dump capacity of the window. Both greedy and randomized algorithms are aided by the propagation rules. As usual in CSP, they allow to avoid inconsistent allocation and to speed up the search. The *Tabu Search* implements an instance of this well known local search procedure for this domain. A specialized move tries to improve the objective function $MTT_\alpha(S)$ performing exchanges on data quantities between pairs of windows w_j.

A detailed description of the problem solving techniques is out of the scope of the present paper. For a more accurate description of MEXAR's algorithmic part see (Oddi *et al.* 2003).

5 The User Interaction with MEXAR

As said in Section 3 MEXAR has been conceived as an interactive decision aid. The problem solving technology has been integrated with interactive functionality that facilitate access to a number of representation and solving functionalities. Indeed the Interaction Module takes into account two main problems:

– *Visualization problem*. To develop trust in the automated solver, a certain level of "transparency" to the user has to be guaranteed. This is done by providing comprehensible and significant representation of the domain model, the problem, its solutions and, to some extent, the problem solving process. Our point here is that to gain user's trust an automated system should be endowed with clear and expressive representation services. We refer to this as to the *glass box* principle, contrasting the widespread trend, among the users, to consider the automated systems as a *black box* of which to be distrustful or suspicious.

– *User participation problem.* The general goal is to capture the different skills that a user and an automated system can apply to the resolution process. Typically an algorithm can perform better on conducting repetitive search steps that are not possible for a human user, while the user usually has more specific knowledge on a domain that is difficult to formalize in general terms to be used by an algorithm. The overall systems Human Planner/Artificial Solver could be considered as more powerful and able to more efficiently solve a problem. The Interaction Module enable such a cooperation and provides interactive services and functionalities to promote a combined problem solving.

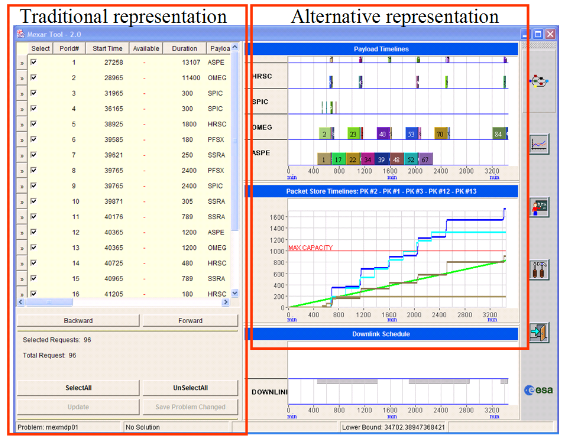
The rest of this section shortly describes the features implemented in MEXAR to address both problems. In (Cortellessa *et al.* 2004) we describe these aspects in more detail and present a preliminary usability study which proved very useful to discover critical interaction aspects as well as to localize the areas in which the whole module needed refinement.

Visualizing domain and problem representation. The model-based approach contained in the problem solving module contains domain features subdivided between activities and resources. Based on a *glass box* principle, MEXAR provides a meaningful visual representation for all the entities relevant in this domain model. Figure 3 presents the basic layout of MEXAR interface. In particular it shows the basic idea used for visualizing a MEX-MDP problem and its solution. In providing visualization facilities within MEXAR we paid attention to design two different interaction modalities, one quite close to the traditional human planner way of work and a second more immediate and intuitive. In this way a user can count on a system that facilitates her task by providing information and solutions close to the way she is used to, and get gradually acquainted with the alternative interaction modalities.

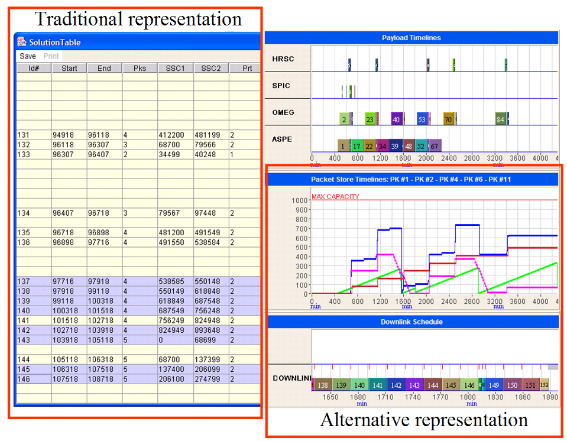
The “traditional representation” of the problem provides a detailed description very close to the one the human mission planner is used to. It shows the list of Payload Operation Requests (PORs) in textual form, that is a detailed list of information on the input activities, their temporal allocation, the distribution of their data on different packet stores. The “alternative” graphic representation provides an additional description of the input activities (PORs) and their distribution on the payloads timelines. Three different panels on the right part of the layout (one on top of the other) represent the timelines (distribution over time axis) of the different domain features relevant for the user: Gantt Chart of PORs in the problem (higher panel on the right); a graphic representation of the temporal function representing the volume of data with the packet store capacity (central panel on the right); the distribution of data dumps on the communication channel (lower panel on the right). These three panels form the basic information the user should be familiar with in order to develop trust in what the solver is representing and managing. A huge software effort has been needed by those three panels that, for example, should be constantly synchronized in their scrolling to represent a consistent model of the world. This temporal representation of the internal symbolic model used by the CSP solver has been instrumental for convincing the ESA personnel that the solver was addressing exactly the problem they had. It is worth noting that the two alternative representations for the PORs (the one on the table and the one on the timelines) are not redundant as they focus on different aspects. In fact, the detailed list of PORs on the left gives a lot of detailed information on the payload activity, while the Gantt on the right focuses on their temporal allocation, and allows to have a feeling of the impact of any POR on the packet stores *via* the other synchronized right panels.

A user can either focus on one or use both the modalities. The two representations are indeed linked and synchronized to each other through a set of interactive links. Alternative representations of the solution can be examined as well: (a) a solution table (traditional representation in Figure 3(b)) is a data structure that reconstructs all the details concerning the solution of the current problem. (b) the alternative graphic view of the solution that is the sequence of dumps on the communication channel (lower right panel). In this last panel a bar represents the time intervals where it is possible to perform memory dumps (visibility windows). Intervals where dumping is not possible are drawn in grey.

It is worth saying that the solution table reflects the usual way of working at ESA, in fact mission planners mainly deal with numerical data contained in spreadsheet tables. Using the table is possible to check for example (a) how data from a single POR is segmented in different dump operations, (b) how the time of data return has been generated, etc. In general, it could be also possible to directly generate



(a) Examining problem features



(b) Studying a solution

Figure 3: Visualization in MEXAR

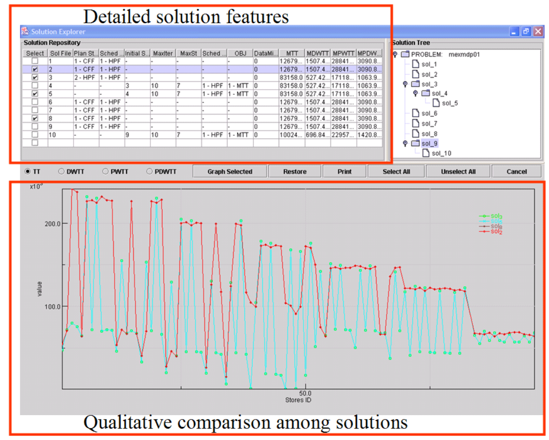
the dump commands from the lines of the table. In fact, the whole table can be saved as a separate file and manipulated by different programs. A number of other small interaction features are implemented as basic functionalities but are not described here because they do not change the general perception of the interaction flow.

User participation in problem solving. Once the human planner has a deeper knowledge of the problem and all the aspects it involves, she can start a different level of interaction with the system trying to contribute with her expertise and judgment to the problem solving. In this way she can possibly choose either to completely entrust the system with the task of finding a solution or participate more interactively in the problem solving process. MEXAR puts at her disposal a second interaction layout, called Solution Explorer, that is intended as an example of this enhanced interaction environment. This second layout has been created mostly for showing ESA personnel an example of advanced functionality based on our interactive problem solving technology.

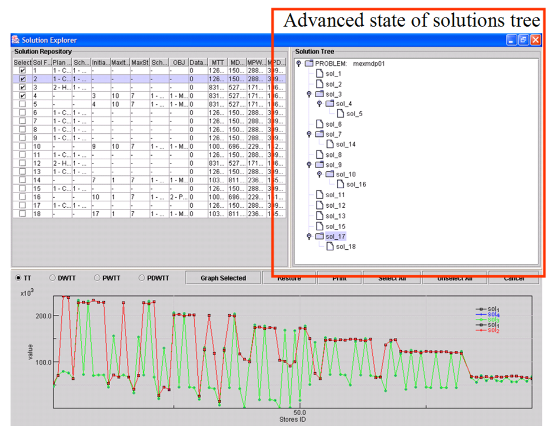
Specific functionalities allow a user to save different solutions for the same problem and guide a search for improvements of the current best result by applying different optimization algorithms. The idea behind this aspect of MEXAR is to involve more deeply the expert user in the problem solving process. A user might generate an initial solution, save it, try to improve it by local search, save the results, try to improve it by local search with different tuning parameters and so on. This procedure can be repeated for different starting points, resulting in the generation of different paths in the search space. Using both the evaluation capability on a single solution and its own experience the user can visit the different solution series, all of them saved, and, at the end, choose the best candidate for execution. Figure 4 contains two examples of use of this interaction environment that refer to a single problem at different stages of exploration.

Studying the examples, it is possible to see that our idea has been again one of facilitating the analysis of the current problem by providing multiple representations of the solutions features. A user has different tools to evaluate the solutions and can either generate new ones or choose the best one according to different temporary criteria. The ideas behind the iterative construction of a solution in MEXAR provides a concept of human guided search (see also different approaches like (Anderson *et al.* 2000)), that can be very useful when solving complex problems. Indeed, in the current version the user uses this functionality as an inspectable repository with a layout that quickly allows for comparison of quality measure of various solutions. Our attention has been devoted to build an interface that allows the user to keep control of the different solution paths and not getting lost in the increasing number of solutions.

This environment represents an interesting experiment of human involvement in problem solving although it has been developed in the context of a limited study which did not have this issue as a primary concern. Nevertheless some interesting functionalities are shown and the possibility is open to create increasingly more useful interactions. In particular it would be worth studying more deeply the competence of the users and experiment how she can be endowed with more strong abilities for influencing the solving strategy of the system through her experience. Among our desiderata is the possibility to continue the development of the system in this direction, not part of the initial study, to obtain a more advanced example of mixed-initiative interaction (Burstein and McDermott 1996; Cohen and et al. 1999).



(a) A step in exploring the solutions space



(b) A more advanced state in the exploration

Figure 4: Involving the user in the solving process

6 Discussion

The study has produced an application that integrates several intelligent techniques to a ground segment mission planning problem. It is worth noting that the effort in MEXAR has been to develop an end-to-end application. In fact the user controls the whole cycle from the problem generation to the solution analysis. This tool is an example of integrated system to endow human mission planners with additional software functionalities that facilitate their work. Several positive aspects are worth mentioning that are important to situate the results in the right context and to appreciate their potentiality:

– The task we have automated usually requires one unit of personnel that is dedicated for half of his working time during the whole mission to manually decide the spacecraft downlink commands. The task of this person is extremely repetitive and involves the control of different details at the same time, mostly represented as numerical values. MEXAR shows a way to support the human operator with a software environment that both decreases the time spent in the task and also requires human cognitive abilities at a higher level of abstraction and decision. In this way, a possibility is open for creating a generation of tools that increases the satisfaction of personnel dedicated to continuous tasks that are critical for space missions success.

– Tools like MEXAR capture an amount of knowledge of the application domain. In particular the CSP internal representation and the algorithms that work on that representation model features of the MARS EXPRESS spacecraft, while the interaction module copes with (and models) aspects of the human work at ground segment. The technology also allows modular extension of such modeling as soon as new features are available or needed. Knowledge preservation opens also the possibility of acceptable turnover of people in charge of the specific tasks during the mission operations.

– MEXAR is a decision aid that preserves the responsibility of the human mission planner. The tool shows how the human user can be supported by tools able to both offer different representation of the problem (e.g., to use reasoning on a graphical representation instead of a numeric one) and to perform combinatorial exploration of alternative solutions (a task difficult for human capabilities). Such features are never aimed at substituting human operators but at supporting them guaranteeing better quality of work. MEXAR is an example of tool that actually empowers humans with additional capabilities through the creation of a cooperative work environment with software tools.

In addition, the technology developed for MEXAR can also be used to produce future tools that address different problems and are used in different phases of a space program. Right now MEXAR is tailored to support the mission planner in deciding memory dumps in a daily activity. The tool can also be used in a preliminary phase of a mission to test, for example, alternative configurations of the on-board memory, payloads and communication channels. In fact MEXAR uses a model of the spacecraft domain that is now integrated by the problem generator to create a MEX-MDP problem specification. It is worth saying that the problem solver is completely parametric with respect to such domain description and it is possible to change such domain definition to simulate different operative scenarios. This feature is suitable to be used in a preliminary phase of a mission for experimenting different policies and mission features.

A further comment concerns the ideas demonstrated in MEXAR about the development of interactive systems for supporting space mission operations. Somehow MEXAR opens a possibility for further investigation of a topic, like the so-called mixed-initiative problem solving, and in general the use of intelligent interfaces integrated with flexible problem solving techniques.

The MEXAR experience is an example of planning and scheduling technology infusion into an ESA space program. Being a study product, the ESA commitment in our work has been quite limited. Nevertheless we have produced a working prototype that can be easily extended to be used in an operational scenario. It is worth remarking that our effort will have a real impact only if the study results will be re-used in different space mission programs. To this purpose it is worth reminding that other ongoing programs now in early stage of development, like for example ROSETTA and VENUS EXPRESS, use a memory management technology that is very similar to MARS EXPRESS. This opens a possibility for early experimenting these techniques on such missions and developing a broad experimentation of the approach. In general, it is our opinion that AI technology is sufficiently advanced and flexible to profitably support human planners in mission planning. In this light, a prompt fusion of expertise between the two communities of AI Researchers and Mission Planning Experts can be fundamental for a truly effective scientific experimentation (see also the recent case of MER (Ai-Chang *et al.* 2004)).

7 Conclusions

In this paper we described our experience in designing and developing a decision support system devoted to solve a complex problem in the context of the MARS EXPRESS space mission.

At the end of this unique experience we can say that providing a useful and effective tool for solving complex problems as MEX-MDP entails a twofold effort: on one side, the development of efficient automated algorithms to rapidly solve the problem; on the other, the design of advanced interactive interfaces that integrate the human operator in the solving process. The need of an interactive user interface is crucial, for mainly two reasons: in all practical cases the user wants to be kept in charge of the decisions taken by the tool; on the other hand, the employed algorithms hardly ever succeed in capturing all the aspects of the real world, which makes constant human supervision and control necessary.

This last important issue pushed us to develop a bipartite architecture whose general design allows it to be used in different contexts. We strongly believe that the effort made in this study, opened the way to a number of new interesting observations which will reveal extremely useful for our future work.

Acknowledgements

The authors would like to thank the ESA-ESOC personnel that enable the MEXAR study introducing them to the MARS EXPRESS world: the project officer Fabienne Delhaise and the MARS EXPRESS mission planners Michel Denis, Pattam Jayaraman, Alan Moorhouse and Erhard Rabenau.

References

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.

D. Anderson, E. Anderson, N.B. Lesh, J.W. Marks, B. Mirtich, D. Ratajczack, and K. Ryall. Human-Guided Simple Search. In *Proceedings of the National Conference on Artificial Intelligence (AAAI 2000)*, 2000.

Mark Burstein and Drew McDermott. Issues in the development of human-computer mixed-initiative planning. In B. Gorayska and J.L. Mey, editors, *Cognitive Technology*, pages 285–303. Elsevier, 1996.

A. Cesta, G. Cortellessa, A. Oddi, N. Policella, and A. Susi. A constraint-based architecture for flexible support to activity scheduling. In *Lecture Notes in Artificial Intelligence, N.2175*. Springer, 2001.

A. Cesta, A. Oddi, G. Cortellessa, and N. Policella. Automating the Generation of Spacecraft Downlink Operations in MARS EXPRESS: Analysis, Algorithms and an Interactive Solution Aid. Technical Report MEXAR-TR-02-10 (Project Final Report), ISTC-CNR [PST], Italian National Research Council, July 2002.

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 - Automating Space Mission Operations using Automated Planning and Scheduling. In *Proceedings of SpaceOps 2000*, 2000.

R. Cohen and et al. What is initiative? In S. Haller, S. McRoy, and A. Kobsa, editors, *Computational Models of Mixed-Initiative Interaction*, pages 171–212. Kluwer Academic Publishers, 1999.

G. Cortellessa, A. Cesta, A. Oddi, and N. Policella. User Interaction with an Automated Solver. The Case of a Mission Planner. *PsychNology Journal*, 2(1):140–162, 2004.

ESA. European Space Agency, MARS EXPRESS Web Site. `http://sci.esa.int/marsexpress/`, 2002.

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

Angelo Oddi and Nicola Policella. A Max-Flow Approach for Improving Robustness in a Spacecraft Downlink Schedule. In *Proceedings of* 4^{th} *International Workshop on Planning and Scheduling for Space (IWPSS-04)*, 2004.

A. Oddi, N. Policella, A. Cesta, and G. Cortellessa. Generating high quality schedules for a spacecraft memory downlink problem. In *Principles and Practice of Constraint Programming, 9th International Conference – CP 2003, Kinsale, Ireland*. LNCS 2833, 2003.

S. F. Smith, D. W. Hildum, and D. R. Crimm. Interactive resource management in the comirem planner. In *IJCAI-03 Workshop on Mixed Initiative Intelligent Systems*, Acapulco, Mexico, August 2003.

E.P.K. Tsang. *Foundation of Constraint Satisfaction*. Academic Press, London and San Diego, CA, 1993.