Proceeding of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space:
i-SAIRAS 2003, NARA, Japan, May 19-23, 2003

An Interactive Problem Solving Tool for Mission Planning *

Amedeo Cesta, Gabriella Cortellessa, Angelo Oddi and Nicola Policella

PST@ISTC-CNR

Planning & Scheduling Team, Institute for Cognitive Science and Technology

National Research Council of Italy, Viale K. Marx 15, I-00137 Rome, Italy

{cesta | corte | oddi | policella}@ip.rm.cnr.it — http://pst.ip.rm.cnr.it

Abstract

This paper gives a short overview of a software system developed to support mission planning tasks in the ESA program called Mars-Express. A system named Mexar is described that synthesizes spacecraft operation commands for downlinking the on-board memory according to mission planners requirements. The tool is based on a constraint-based representation, a family of problem solving algorithms and a sophisticated user front-end that allows to understand and manipulate different features of both the current problem and the proposed solution.

Figure 1: Artist's impression of Mars-Express in orbit around Mars (Courtesy of ESA).

1 Introduction

The Pst group at ISTC-CNR has a long experience in research and development of intelligent techniques for planning and scheduling (P&S) (e.g., [8; 5; 2]). In the period November 2000 - July 2002, the group has been working to the ESA study "Efficient Planning Algorithms for an Interplanetary Mission". The study aimed at demonstrating Artificial Intelligence techniques for P&S in a real space problem, the mission planning for Mars-Express [9]. Mars-Express is a space probe, see the sketchy view in Figure 1, that is going to be launched during 2003 and will be orbiting around Mars during 2004 and 2005.

In May 2002, the Pst has delivered to ESA-ESOC the final version of a software system called Mexar that is able to automate the generation of spacecraft operations to downlink the on-board mass memory. The results are described in the final report delivered in July 2002 [6]. 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 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.

This paper is aimed at describing the main aspects of Mexar and to give some comments on the experience. It is organized as follows: Section 2 defines the specific mission planning problem, Section 3 introduces a CSP approach, and a software architecture for such an approach. The section gives also an overview of the problem solving methods. Section 4 specifically presents Mexar as an interactive system that empowers users with additional solving capabilities for mission planning. A concluding section summarizes and reasons about the main achievements of the project.

2 Problem Description

Inside the complex domain of a space mission like Mars-Express we have been concerned of data transmission to Earth. A space probe continuously produces a large amount of data. Because Mars-Express is a single pointing probe, it points either to Mars (data production) or to the ground (data downlink). As a consequence data (from both scientific observations and from house-keeping) are first collected in the finite capacity on-board memory. The problem solving goal consists in

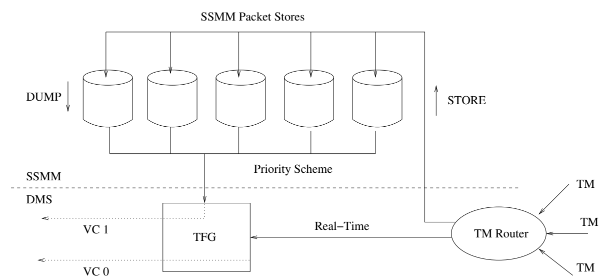

Figure 2: On-board telemetry flow. The different telemetry (TM) data produced on board are stored on the on board memory (SSMM) subdivided in packet stores. Memory stores are then downloaded with different dumps that transfer data to the ground.

synthesizing spacecraft operations for emptying as much as possible the on-board memory during downlink time intervals, to allow the spacecraft to save new information without losing previous data.

Different constraints are conflicting with the aim above. Besides the communication channel availability there are different transmission rates to be taken into account. Additional constraints rise from the specific use of the on-board memory. That memory, indeed, is split in different memory banks (or packet stores) each of them having a finite capacity. For each piece of information produced inside the probe, a packet store is also defined in which such data should be stored. Different data are stored in a sequential way and the packet stores are managed cyclically. As a consequence, in case the memory is full, the stored data are lost when new data become available.

We have formalized this problem as the Mars Express Memory Dumping Problem (Mex-Mdp) whose different components are described in the rest of this section.

2.1 The Memory Dumping Problem

The basic ontological objects of the Mex-Mdp domain are *resources* or *activities*: resources represent domain subsystems able to give services; activities model tasks to be executed using resources over time. A set of *constraints* defines needed relationships between the two type of objects. Figure 2 shows a sketch of the Mars-Express modules that are relevant to Mex-Mdp. The rest of the section provides definitions of the domain features, and describes the constraints contained in the problem.

The relevant types of resources in Mex-Mdp are the set of packet stores, the set of on-board payloads and the set of communication channels:

– *Packet Stores.* The on-board memory (Solid State Mass Memory or SSMM) is subdivided in a set of separated packet stores pk_i, i.e. they cannot exchange data among them. Each one has a fixed capacity c_i and a its own priority value, p_i. Each packet store can be seen as a file of a given maximal size that is managed cyclically. Data are organized in data packets and then stored in the assigned packet store.

– *On-Board Payloads.* An on-board payload can be considered as a *finite state machine* such that, each state has a different behavior in generating observation data. In particular to 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 (it will be 0 in case of no-transmission window).

The amount of each resource is constant and known in advance.

Activities describe *how* resources can be used. Three types are relevant in Mex-Mdp: payload operations, memory dumps and continuous data streams. Each activity a_i has an associated execution interval, which is identified by its start time $s(a_i)$ and end time $e(a_i)$. Each type of activity is characterized by a particular set of resource requirements and constraints.

– *Payload Operations.* A payload operation por_i corresponds to scientific observation. Each observation generates an amount of data that, according to the Mars Express operational modalities, are decomposed in different *store* operations, and distributed over the set of available packet stores.

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

– *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.

2.2 MEX-MDP

Having defined the basic domain entities we define now the Mex-Mdp. Given a set of scientific observations $POR = \{por_1, por_2, \ldots, por_n\}$ and a domain description that specify all the previously introduced constraints, a *solution* is a set of dumping operations $S = \{md_1, md_2, \ldots, md_s\}$ such that:

– The whole set of data are "available" on ground within the considered temporal horizon $\mathcal{H} = [0, H]$.

– Each dump operation starts after the generation of the corresponding data. For each packet store, the data are 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 is below or equal to its capacity c_i (not overwriting is allowed).

Even if a solution satisfies all the imposed constraints, it could be not sufficient yet. A main goal is to figure out *high quality solutions* with respect to a set of evaluation parameters: a high quality plan delivers all the stored data as soon as possible according to a definite policy or objective function.

Two relevant pieces of information to build up an objective function are:

1. 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 its end time;
2. the priority values of the data generated by the por_i.

The objective function used is the *mean* α*-weighted turnover time* MTT_α of a solution S:

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

Given an instance of a MEX-MDP, an *optimal solution* with respect to a weight α is a solution S which minimize the objective function $MTT_\alpha(S)$.

According to ESA requests, we consider the *Mean Turnover Time* (MTT) (with $\alpha_i = 1$, $i = 1..n$) as an evaluation metric for good solutions.

3 A CSP Approach to the Problem

The approach used in solving MEX-MDP has been influences by PST previous work in the field of planning and scheduling. We used modules of our open architecture O-OSCAR to represent, solve and present to the user the problem and its solutions [8; 4]. The work on O-OSCAR has been devoted to the development of a standard framework able to capture different aspects of a scheduling domain.

Artificial Intelligence techniques work on a symbolic model of the domain of application. CSP problem solving is based on a powerful modeling capability that allows to represent a large number of real domains. O-OSCAR is based on the CSP modeling ability integrated with various modules for managing different aspects that develop a complete scheduling system. In this section we summarize the main modules of this kind of architecture and explain how it has been enriched and empowered in

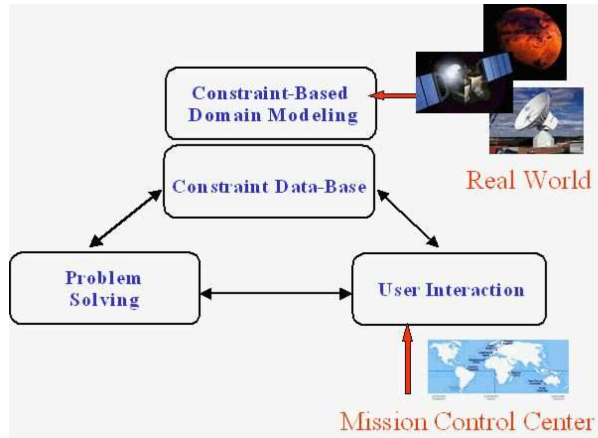

Figure 3: A CSP architecture

order to address the MEX-MDP problem evolving in a different CSP-based architecture, named MEXAR.

A CSP-based software architecture can be summarized as in figure 3. Four main modules are integrated to create a complete approach to a problem:

Constraint-Based Domain Modeler: This is the module that allows to capture and model the relevant features of the real domain. It guarantees a representation of the main components of the domain, and the dynamic rules according to which those components evolve. This module in MEXAR allows to describe the different features of a MEX-MDP formalized as a pure CSP (see [6]).

Constraint Data Base: The central part of a CSP architecture is the Constraint Data Base (CDB), a module all the other ones rely on. It provides a set of data structures able to represent constraints and guarantee a set of basic functionalities for their management.

Problem Solver: This module is responsible for implementing the solution algorithms. The idea behind this module is the one to provide a user with a portfolio of algorithms enabling him to choose among different strategies tuning the parameters according to his/her preferences. In a planning and scheduling domain, very often problems cannot be solved to optimality, so good approximations of the optimal solution are searched for. A solutions is then the result of an iterative process: one or more initial schedules are found, problems in these schedules are identified, constraints are relaxed and changes are made on the solution. We allowed an iterative process endowing the user to use multiple algorithms he can choose from, like a local search procedure or different sampling algorithms.

User Interaction Environment: An approach based on solution exploration requires support of a robust interaction layer that keeps the user in the loop. This module directly interacts with the user, to allow him not only to use the tool but also to take part in the advanced problem solving functionalities that involve him into the solving process. This module is very important because represents the communication channel between the user and the automated solver. We believe that it should guarantee a productive and profitable interaction able to enhance the problem solving capabilities. The idea 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 User Interaction Environment plays a crucial role in enabling such a cooperation.

Mexar and O-Oscar share the same development methodology sketched in Figure 3. In Mexar the CDB and the solving algorithms are tightly coupled (see the short description in 3.1) and the Interaction Environment in quite accurate (Section 4).

3.1 Problem Solving

Our approach to solving Mex-Mdps is grounded on the CSP paradigm [12]. The problem is represented by a set of objects or variables and a set of constraints that define the relations among them. A solution consists in assigning to each variable one of its possible values such that all the constraints are satisfied. The search for a solution to a CSP can be viewed as modifying the current (partial solution) by addition and removal of constraints. Very important in a CSP paradigm is the use of *propagation rules*. Through them it is possible to examine the existing search state to find implied commitments. Though, in general, it is not possible to remove all inconsistent values through propagation rules, they can speed up the solver procedure.

In modeling Mex-Mdp we choose as variables the amount of data that in each temporal interval is dumped from each packet store. On the other side the input are the flow of data stored for each packet store. Two type of constraints are presented: (1) the *conservative constraints* and (2) the *downlink constraints*. The former claims that the difference between the amount of generated data and the amount of dumped data cannot exceed both the capacity of the packet store capacity and the generated data (*overdumping*). The latter states that is not possible to use the channel more than its availability.

The problem solving approach is based on two levels of abstractions that follow the previous representation:

- *Data Dump Level.* In that level a solution consists to figure out an assignment for the set of variables such that the constraints of Mex-Mdp are satisfied. Below we describe three different approaches to find a possible solution.
- *Packetization Level.* In this phase the memory dump operations are generated according to the solution found in the above level. In other words the solution of the previous phase is translated in a set of dump activities. For each memory dump operation the scientific observation(s) or the house keeping track related to, are computed.

Three Solving Algorithms. On top of the constraint based representation we have synthesized different solving algorithms for Mex-Mdps.

Greedy Algorithm: That basic approach consists of assigning a value to each variable according to a heuristic. The variable have been selected considering an increasing temporal order. Two different solving priority rules are implemented:

- *CFF* (Closest to Fill First) that selects the packet store with the highest percentage of data volume.
- *HPF* (Highest Priority First) that selects the packet store with the highest priority. In case that a subset of packet stores has the same priority, the packet store with the smallest store as outcome data is chosen.

An important aspect is represented by the use of the propagation rules that allow to avoid some inconsistent allocation and to speed up the search (this is also true for the randomized approach).

Randomized Algorithm: Random search [13] is a powerful and widely used solving technique and its joint use with Constraint Reasoning has been successfully used in several kind of problems (for scheduling problems see [14; 7] as examples). 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 present in the packet store, (2) the amount of data already planned to be dumped and (3) the dumping capacity of the window.

Tabu Search: Briefly the approach consists of changing incrementally the start solution to achieve a better one according to an objective function. Each time a better solution is found the process restarts using that one as the "new" starting point. In our implementation the process terminates as soon

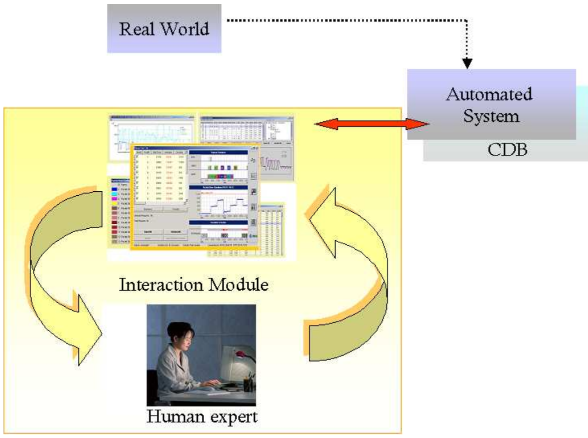

Figure 4: An integrated ⟨human-computer⟩ solver

Figure 5: The Problem Analyzer Layout

as a certain number of changes without any improvement is done. A fundamental feature of tabu approach is a short-term memory that, storing the last moves done, allows to avoid loops (for more details about tabu search and how we implemented it the reader can see respectively [10; 11] and [6]).

4 Enhancing Human Problem Solving Capabilities with MEXAR

In developing a support tool for MEX-MDP the pursued idea has been of endowing a human planner with a sophisticated and helpful tool able to support and enhance his/her solving capabilities. Such a tool should guarantee a user to more easily inspect a problem, analyze its features, find a satisfactory solutions as a step by step procedure, having continuous control over the problem solving process. The User Interaction Environment should also guarantee an easy interaction protocol, providing friendly representations of the problem, the solutions and all the entities of the domain, while hiding the underlying complexity to the user so that he could concentrate his energy to more high level decision tasks. In this way a user can count on a sort of "external prosthesis" of his problem solving capabilities entrusting the system with tasks either repetitive or boring, or requiring the management of large amount of data or a strong computational activity. We consider the artificial system as an amplifier of the human problem solving capabilities and the integrated system "human-computer" (see figure 4) as very powerful and able to capture the different and complementary skills of both the human and the artificial solvers. In such a vision the solution to a complex problem is obtained as result of an iterative and interactive process which both these entities can contribute to.

4.1 Different Levels of Interaction

The MEXAR interaction module has been designed subdividing the interactive capabilities into 2 levels. A first set of facilities provides a user with the possibility to load a problem, inspect features according to different and alternative perspectives, and to specialized interactive tools. It is also possible to ask for the initial solution to the problem and examine and evaluate different aspects of it (e.g. quality, correctness). We grouped together this facilities in a unique graphic environment named Problem Analyzer (PA). Then a more advanced interactive environment has been built that offers a deeper level of interaction. This is what we call Solution Explorer (SE) that will be described in 4.3. This subdivision of interaction levels is to provide different services: the first level allows a user to get acquainted with the problem and its features, having however the possibility to obtain a first solution to it. Once the human planner develops a deeper knowledge of the problem and all the aspects involved in, he/she can start also the second level of interaction trying to contribute with his/her expertise and judgment to the problem solving. Different users and even the same user at different time could desire to interact with the system according to different levels of interaction, choosing either to completely entrusting the system with the task of finding a solution or to participate more interactively in the problem solving.

4.2 The Problem Analyzer

The PA contains two groups of functionalities one for problem editing and refining, another for problem solving. The interaction layout is based on the idea of inspectability of all the different components of the representation of the MEX-MDP problem. This inspectability follows what we call "glass box principle" [3]. In fact, it allows the user to visually control the temporal behavior of the domain variable that are modeled, somehow "seeing" the internal symbolic representation of MEXAR.

The example in Figure 5 shows the basic idea used for visualizing a MEX-MDP problem and its solution. A specific dialogue allows to choose one of the problem files (see also [6]). The CSP representation for the problem is instantiated and used for showing information on the interaction panel. It basically represents different aspects of the problem: the Payload Operation Requests (PORs) list in textual form (left panel in the layout); their distribution over the payloads timelines that is a Gantt Chart of the problem (higher pane on the right); a graphic representation of the temporal functionl representing the volume of data with the packet store capacity (central panel on the right). Additionally the user can choose between the different problem solving strategies from the pop up menu bar on the right.

After the solver work a solutions is displayed: the communication downlink window is shown in the lower panel on the right and shows the solution as a sequence of dump operations. A bar representing the time intervals where it is possible to perform memory dumps (visibility windows) is shown right below the line delimiting the channel. Intervals where dumping is not possible will be drawn in grey.

We can see in the figure that the maximum capacity constraint is never violated due to the effect of different dumps (the packet store profile is under the max capacity). In order to guarantee different perspective of the solution we also provided a solution table that is a data structure that reconstructs all the details concerning the solution of the current problem. It is worth saying that the solution table reflects the current 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 the data from a single POR are 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 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.

Another feature of the module is the possibility to evaluate a solution according to some metrics. This possibility allows the user to easily estimate the quality of a solution with respect to some chosen parameters. In fact, a graphic evaluation has been added to obtain an immediate level of evaluation of the current solution. When calling the command from the menu a dialogue window allows to choose one or more evaluation functions (e.g. Turnover Time, Data Weighted Turnover Time, etc.).

4.3 The Solution Explorer

As we said before our idea has been to give a user the possibility to apply different solving methods to the same problem (greedy solver, randomized algorithm, local search, each with different possible tuning parame-

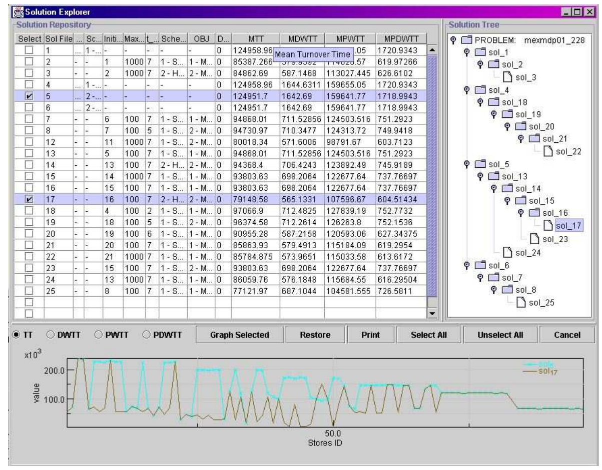

Figure 6: The Solution Explorer Layout

ters). This capability suggested us to actively involve the user in the choice of which algorithm to apply. Specific functionalities allow the user to save different solutions for the same problem. In addition, the user can guide a search for improvements of the current best result applying different optimization algorithms. At this aim we developed the Solution Explorer (SE) which provides an additional level of interaction. The idea behind the SE is that an expert user could try to participate more deeply in the problem solving process. A user can 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. In this way, it is possible to generate paths in the search space. The user can restore one of the previous solutions and try to improve it with a local search with different parameters, etc. In this way he generates a tree of solutions. This procedure can be repeated for different starting points generating in this way a set of trees (see Figure 6). Using at the same time the evaluation capability on a single solution and its own experience the user can generate different solution series, all of them saved, and, at the end, choose the best candidate for execution. The SE contains three panels: a table containing a distinct saved solution on each line and the information on how it has been generated (with greedy, randomized or with taboo) and the average evaluation parameters for that solution; a simple representation of the set of solution trees that are generated for that problem; a graphic window where it is possible to compare solutions among them according to different parameters.

Again the idea is the one of facilitating the analysis of the current problem by the user. He has different tools to evaluate the solutions and can either generate new ones or choose the best according to different temporary criteria. In this case the idea is that the expertise of a human planner could influence the search in the space of solutions so that from an initial solution other better ones can be obtained.

5 Discussion

The Mexar study has been able to produce results at three distinct levels that we summarize here separately.

Problem Study: The open problem that has been jointly identified by the Pst members, Project officer and Mars-Express MPS experts. The task has been studied and formalized as the Mars Express Memory Dump Problem (Mex-Mdp). In addition, the Mex-Mdp properties have been analyzed to synthesize: (a) a lower bound for the more frequent evaluation measure (the Mean Turnover Time (MTT)); (b) a set of macro evaluation parameters for characterizing the difficulty of a problem. Additionally, a random problem generator, grounded on the MPS experience, has been synthesized that helps in the definition of a "reasoned" set of benchmarks.

Problem Solution: This is the critical result we have obtained. We have been actually able to automatically solve Mex-Mdps using constraint-satisfaction (CSP) techniques. On top of a CSP representation of the problem, several solving techniques have been designed and implemented to work in an integrated way. In particular specific results concern: (a) basic deduction rules that are grounded on the specific problem and help pruning the search space; (b) a set of solution procedures that range from a basic greedy procedure, to an iterative sampling schema, to a tabu search meta-heuristic.

Tool Development: The solution grounded on constraint management has been the main result of our study, but we have also pursued a second ambitious goal: to develop a technological demonstrator able to show how these techniques could open possibilities for intelligent support to human mission planners. This second goal has also required a huge investment in human resources but has been important for us to show the technology at work. We have build Mexar, a Java application that interacts with users to solve different Mex-Mdps. The system is endowed with a quite sophisticated interaction module that pursues the idea of supporting the users by preserving their complete control on the critical decisions.

Mexar has been successfully delivered to the MPS experts that are actually responsible for solving Mex-Mdps during Mars-Express orbiting around Mars. Such users have had a key role in validating the work at different steps of the study. They have been also very relevant to clarify aspects of the problem first, and to suggest ways for improving several aspects of the systems. In fact, the current release of Mexar derives from two preliminary versions that integrated simplified version of both the solver and the interaction modules.

It is worth underscoring the huge effort that has been done with respect to the basic goal of solving the problem. The Pst members are particularly aware that a real impact of intelligent problem solving technology is achieved only if the research effort is integrated in a usable environment that takes into account all the problems of usability and control by human users.

5.1 Concluding Comments

The study produced an application that integrates Artificial Intelligence techniques to a ground segment mission planning problem. It is to be noted that the effort in Mexar has been to develop an end-to-end application that allows the user to control the whole cycle from the problem generation to the solution analysis. Several positive aspects are worth mentioning that are important to situate the results in the right context and to appreciate its 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 increase 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 a generation of 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 to support 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.

– A different comment concerns the use of the MEXAR tool. It is right now tailored to support the mission planner in deciding memory dumps in a daily activity. It is worth noting that 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 of the mission features.

– A final 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. This experience may represent a step that can have an impact of different space mission programs (see [1] for another example).

Acknowledgements. This work could not be possible without the support at ESA-ESOC of both the project officer Fabienne Delhaise and the MARS-EXPRESS mission planners Michel Denis, Pattam Jayaraman, Alan Moorhouse and Erhard Rabenau.

References

[1] M. Ai-Chang, J. Bresina, L. Charest, A. Jonsson, J. Hsu, B. Kanefsky, P. Maldague, P. Morris, K. Rajan, and J. Yglesias. MAPGEN: Mixed Initiative Planning and Scheduling for the Mars 03 MER Mission. In *Proceeedings Seventh Int. Symposium on Artificial Intelligence, Robotics and Automation in Space (i-Sairas-03), Nara, Japan, May*, 2001. This proceedings.

[2] L. Carlucci Aiello, A. Cesta, E. Giunchiglia, and P. Traverso. Merging Planning and Verification Techniques for "Safe" Planning in Space Robotics. In *Proceeedings Sixth Int. Symposium on Artificial Intelligence, Robotics and Automation in Space (i-Sairas-01), Montreal, Canada, June*, 2001.

[3] A. Cesta, G. Cortellessa, A. Oddi, and N. Policella. Interaction Services for Mission Planning in MARS EXPRESS. In *Proceedings of the 3rd International NASA Workshop on Planning and Scheduling for Space, Houston, Texas, October 27-29*, 2002.

[4] A. Cesta, G. Cortellessa, A. Oddi, N. Policella, and A. Susi. A Constraint-Based Architecture for Flexible Support to Activity Scheduling. In *in AI*IA 2001: Advances in A. I., LNAI 2175*, 2001.

[5] A. Cesta, G. Cortellessa, A. Oddi, and A. Susi. Synthesis and Execution of Space Mission Plans with the O-OSCAR Software Architecture. In *Proceeedings Sixth Int. Symposium on Artificial Intelligence, Robotics and Automation in Space (i-Sairas-01), Montreal, Canada, June*, 2001.

[6] 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.

[7] A. Cesta, A. Oddi, and S. Smith. A Constrained-Based Method for Project Scheduling with Time Windows. *Journal of Heuristics*, 8(1):109–135, 2002.

[8] A. Cesta, A. Oddi, and A. Susi. O-OSCAR: A Flexible Object-Oriented Architecture for Schedule Management in Space Applications. In *Proceedings of the Fifth Int. Symp. on Artificial Intelligence, Robotics and Automation in Space (i-Sairas-99)*, 1999.

[9] E. S. A. (ESA). MARS EXPRESS Mission Web Site. http://sci.esa.int/marsexpress/, 2002.

[10] F. Glover. Tabu Search – Part I. *ORSA Journal of Computing*, 1:190–206, 1989.

[11] F. Glover. Tabu Search – Part II. *ORSA Journal of Computing*, 2:4–32, 1990.

[12] U. Montanari. Networks of Constraints: Fundamental Properties and Applications to Picture Processing. *Information Sciences*, 7:95–132, 1974.

[13] S. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[14] A. Oddi and S. Smith. Stochastic Procedures for Generating Feasible Schedules. In *Proceedings 14th National Conference on AI (AAAI-97)*, 1997.