ARTICLE

Fast Production and Flexible Maintenance of Schedule for Space Applications

Authors: **A. Cesta, A. Oddi, and A. Susi**

This paper presents an overview of a software system named O-OSCAR for schedule production and management in space applications. The main features of the architecture are presented and some aspects of the current application to project management are discussed.

Keywords: scheduling, resource management, planning, space applications, interactive support systems.

Introduction

This paper describes research work supported by the Italian Space Agency (ASI) aimed at realizing a software architecture for planning and scheduling in space projects [2]. The research goal has been the definition of a software framework to be re-used in a class of similar problems sharing requirements typically holding in space applications.

We have addressed a class of scheduling problems that involve quite complex time and resource constraints. Time constraints represent for examples set-up times for instruments, target visibility windows, transmission times (e.g., to represent memory dump), deadline constraints, etc. Resource constraints can represent capacity of on board memory (e.g., tape recorder or solid state recorder capacity), transmission channel capacity, and energy bounds (e.g., limitation on the number of active instrument in a space probe). From the scheduling literature point of view a reference problem can be the so-called Resource Constrained Project Scheduling Problem with Generalized Precedence Relations (RCPSP/max) [5, 3]. In such a problem a network of activities is to be executed on a number of resources having capacity greater than 1. Activities should satisfy reciprocal temporal constraints specifying minimum and maximum separation slacks. Specific attention has been given to create software functionalities particularly useful in space: (a) since space missions span for several years, a major role assumes the possibility of modifying plans and schedules, as well as the details of the application domain, as soon as the steps of a mission become more mature; (b) the ability of quick schedule production has been pursued by allowing the integration of different solution methods on top of a module that represents the domain features and constraints; (c) the explicit consideration given to aspects of user interaction and acceptance of the automated system in different working environment (both on ground and on board segments).

The major result of our work has been the software architecture named O-OSCAR (Object-Oriented SCheduling ARchitecture) that represents a carefully designed library of functionalities to support the previous requirements in an integrated way. This paper presents the main ideas contained in O-OSCAR[2] by showing its main functionalities and illustrates some details of a complete system for project scheduling we have realized.

Basic O-OSCAR Components

In developing a complete solution to a planning/scheduling problem several basic aspects need to be integrated. Figure 1 shows the different modules of O-OSCAR. The dotted and numbered boxes identify three major components:

[2] Current information about the O-OSCAR system are available at `http://pst.ip.rm.cnr.it/`

(1) the module that represents the problem domain and the problem solution; (2) the module responsible for problem solving decision; (3) the module that takes care of the interaction with the user.

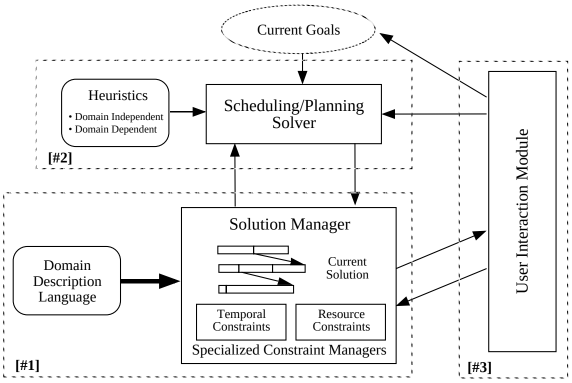

Figure 1. O-OSCAR Components

The module labeled #1 plays the key role in the application of AI constraint-based methodology. This module is in charge of two strictly interconnected aspects:

Domain Representation. A Domain Representation Language allows the system developer to describe different aspects of the world that the scheduling system needs to know in order to produce a solution. Usually such languages represent a class of problems by defining their main objects and their peculiar constraints.

Solution Representation and Management. The constraint-based approach to scheduling is centered on the production and maintenance of a solution. Such solution consists of a representation designed on top of specialized constraint reasoners. The constraint model represents particular aspects of the domain (e.g., temporal features, resource availability) and is called into play when changes are performed by a problem solver or a user. The solution manager is usually endowed with a set of primitives for communicate changes and formulate queries. The solution manager is the core of the constraint-based approach, it offers an active service [1, 4] that automatically takes care of checking/maintaining the satisfaction of the basic domain constraints. Once realized this module, a complete approach to the solution is obtained addressing the two missing aspects:

Automated Problem Solving. Two features are needed: (a) the definition of an open framework to perform the search for a solution; (b) the identification and representation of heuristic knowledge to guide search for avoiding computational burden (the two components in box labeled #2 of Figure 1). The Planning/Scheduling Solver identifies the software component that can search for a solution using a portfolio of different methods (e.g., exhaustive search procedures [5], greedy heuristics [3], local search approaches). The rounded box “Heuristics” underscore a representation task for this module to store specific problem solving knowledge used during search.

User-System Interaction. This module (box #3) allows the interaction of the user with both the solution and the problem solving methods. The interaction functionalities may vary from more or less sophisticated visualization services, to a set of complex manipulation functionalities on the solution allowed to the user. This aspect, usually neglected in problem solving systems, is crucial to develop effective applications because the acceptability of an innovative tool is strictly connected with its usability and intuitiveness.

The basic working cycle of the architecture starts from the user that defines a problem to be solved and eventually communicates to the solver specific goals to be satisfied on this problem (e.g., choice of a particular solution strategy or definition of an evaluation metric for the solution). The problem is represented in the solution manager and the solver starts searching for a solution by querying the manager for specific information and communicating it the modifications to register in the partial solution to avoid conflicts in the satisfaction of the various constraints. When a complete solution has been found the user can inspect it and investigate several specific aspects (e.g., current available resources, etc.). A further level of functionalities is still available to the user to update the problem definition when a solution to the previous problem already exists, and to ask the problem solver to accommodate changes to existing schedule due to modifications of the external scenario (e.g., sudden activities needed for a repair command that were not included in the initial problem).

Conclusion

We have described O-Oscar an open architecture for scheduling and schedule management of rather complex problems. O-Oscar represents a modern constraint-based approach to schedule management from the resolution of a new instance of a problem to the management of changes arising during schedule life-cycle. A continuous attention has been dedicated to the investigation of human-computer interaction aspects customized to the application domain. We are currently working at a real space problem that requires to customize O-Oscar for obtaining a decision support tool.

Acknowledgments. This research has been developed in the framework of the project “A toolkit for the synthesis of interactive planners for complex space systems” supported by ASI (Italian Space Agency) under contract ASI-ARS-99-96.

Bibliography

[1] R. Cervoni, A. Cesta, and A. Oddi. Managing Dynamic Temporal Constraint Networks. In *Artificial Intelligence Planning Systems: Proceedings of the Second International Conference (AIPS-94)*, 1994.

[2] A. Cesta, A. Oddi, and A. Susi. O-Oscar: A Flexible Object-Oriented Architecture for Schedule Management in Space Applications. In *Proceedings of the* 5th *International Symposium on Artificial Intelligence Robotics and Automation in Space (i-SAIRAS'99)*, 1-3 June 1999 (ESA SP-440).

[3] A. Cesta, A. Oddi, and S.F. Smith. An Iterative Sampling Procedure for Resource Constrained Project Scheduling with Time Windows. In *Proceedings of the* 16th *Int. Joint Conference on Artificial Intelligence (IJCAI-99)*, 1999.

[4] A. Cesta and C. Stella. A Time and Resource Problem for Planning Architectures. In *Proceedings of the Fourth European Conference on Planning (ECP-97)*, 1997.

[5] B. De Reyck and W. Herroelen. A Branch-and-Bound Procedure for the Resource-Constrained Project Scheduling Problem with Generalized Precedence Relations. *European Journal of Operations Research*, 111(1):152–174, 1998.

Author information:

Amedeo Cesta, Angelo Oddi, and Angelo Susi
IP-CNR, National Research Council of Italy, Viale Marx 15, I-00137 Rome, Italy, {cesta, oddi, susi}@pst.ip.rm.cnr.it □