POSITION STATEMENT. InProceedings of the IJCAI2001. Working notes of the IJCAI'01 Workshop on Planning with Resources. IJCAI Press, Seattle, WA (USA), August 2001, pp 66-67.

APPLYING AN AI PLANNER TO SCHEDULE GROUND SATELLITES OPERATIONS

MD R-Moreno[1], D. Borrajo[2], D. Meziat[1]

[1]Departamento de Automática. Universidad de Alcalá (Madrid). Spain

{mdolores,meziat}@aut.uah.es

[2]Departamento de Informática. Universidad Carlos III de Madrid (Madrid). Spain

dborrajo@ia.uc3m.es

Introduction

Planning and scheduling are closely related areas that AI community has been long concerned with. The first one deals with finding plans to achieve some goals from an initial state. The second one refers to the allocation of available resources to known activities over time in order to produce schedules that respect temporal relations and resource capacity constrains. They can also optimize a set of objectives, such as minimize tardiness, minimize work in process, maximize resource allocation or minimize cycle time. Planning refers to what should be done and scheduling to how it should be done. Due to these differences, planning and scheduling have been traditionally solved separately using rather different methods.

Depending on the problem complexity some domains allow a strict separation between planning and scheduling. But, in other cases, there is an indirect temporal and resource dependency with other states and goals that can not be taken into account if we separate both tasks. In this paper we present a way of integrating them that consists on allowing the planner to reason about resources, although not as explicitly as in the case of other systems. We wanted to explore how to use a standard planner, PRODIGY, to also schedule in a real domain. In this approach, the designer has to be able of determining the elements that can and can not be tackled and the way to express them in the planner language for an efficient internal treatment.

In PRODIGY there is not a language that allows an explicit representation of resource or temporal information as in [Chien et al 2000]. However, thanks to three representation facts, the user can use numeric variables and constrains. They are: its capability to represent infinite types (numeric variables); the possibility to define functions that obtain variables values in preconditions of operators; and the use of control rules to prune the search.

We have used this approach to plan and schedule nominal operations to perform in three satellites along the year for a Spanish satellites company, HISPASAT.

Satellite Operations

This section gives a brief overview of the HISPASAT ground scheduling operations for its three satellites. HISPASAT is a Spanish multi-mission system in charge of satisfying national communication needs, and supplying capacity for digital TV in Europe and America, TV image, radio and other signals, and special communications for defense purposes. Every operation in orbit must have explicit engineering instructions. An operations engineering group generates this documentation every year *by hand* and *in paper*. Later, this documentation is revised and verified. Due to the increasing number of satellites, actually three: 1A, 1B and 1C and shortly the 1D satellite, a program that handles and validates the operations for engineering support is needed.

Scheduling the nominal operations for on orbit control can be viewed as a constraint satisfaction problem where each satellite has its own restrictions depending on the different events that can occur over time. We could identify three categories of planned operations, according to the flexibility to schedule them:

- Operations which are driven by external events and should start or be completed at a fixed time. Examples are Moon Blindings, Sun Blindings or Eclipses of the Sun by the Earth or by the Moon.
- Operations which are part of a determined strategy and should be performed at given dates, though some flexibility exists to modify the strategy. This is the case of Maneuvers, Localization Campaigns or Batteries Reconditioning.
- Operations that should be performed within a wide period of time and a significant flexibility exists for scheduling as Steerable Antenna Maintenance.

The scheduling is done in a year basis, although a weekly and daily schedule is produced in case of changes in some operations. It is the case of the need to point the antennas to a particular direction following the customer demands.

The Planning and Scheduling Model

PRODIGY is an integrated intelligent architecture that has been used in a wide variety of domains [Veloso et al 1995]. The problem solver uses a backward chaining means-ends analysis search procedure with full subgoal interleaving. The planning process starts from the goals and adds operators to the plan until all the goals are satisfied. The obtained plan does not consider any optimization with respect to resource use or availability, given that for HISPASAT it is enough to find a plan. However, the planner could obtain an optimal plan according to some criteria using the QPRODIGY version described in [Borrajo et al. 2001].

Three types of knowledge are the inputs to the planner. First, the domain theory that contains all the actions represented by so-called operators. Each operator in the HISPASAT domain represents an action to be taken by a satellite on orbit. An operator consists of pre-conditions (conditions that must be true to allow the action execution), and post-conditions or effects (usually constituted of an add list and a delete list). The add list specifies the set of formulae that are true in the resulting state while the delete list specifies the set of formulae that are no longer true and must be deleted from the description of the state. Also, all operators have as variables, at least, the satellite, and the start and end times of the corresponding action. In some of the operations, resources as the fuel and oxidant tank, the gyroscope or the branch in which the maneuvers are performed are also specified.

To represent the dates in the operators, we have chosen the "seconds since 1900 in GMT time zone" representation (we have used Common Lisp as the programming language for functions given that PRODIGY is written in Lisp). The reason to use this format is for efficiency: it will be faster to generate the bindings for one variable than for the six usual time dependent variables (corresponding to the year, month, day, hours, minutes and seconds) and also GMT is the reference zone time for HISPASAT. Other approaches fix the starting point of the computation, call it 0, and schedule from that point on all the activities.

Thanks to the ability to code functions, we can add constrains within and among operations, such as release times and duration to the operators. For instance, to handle duration of operations, we have coded functions that calculate the new time from the start time and the expected duration. In the effects list of the operator we add a predicate that contains this new time (i.e., the end of the operation). Another example are the constrains among operations. If another *operation-B* starts two hours after the end of *operation-A*, the *operation-B* operator will have as one precondition the predicate that was added in the effects of *operation-A*. If it is a range of values, the *operation-A* operator will generate all values between the range limits. Since those are continuous variables, only discrete values are generated. Since, in this domain, there are no tight constrains among operations, this was a feasible solution. The ability to handle resources with variable bounds is solved using the same scheme.

The second input to any planner is the problem, described in terms of an initial state and goal. The initial state describes all the events that will occur during the year related to the satellites operations, such as expected moon blindings or eclipses, seasonal events such as equinox or solstice or unforeseen events such as the Leonids. These dates are represented, as well, using the seconds representation. With respect to the goals, they refer to the twenty-seven maneuvers that have to be performed in each satellite along the year, boost heating (using determined tanks), localization campaigns, battery reconditioning, etc. For each satellite there are 220 goals and 100 initial conditions. The average number of operators in the plan for each satellite is 450. It takes less than five minutes to generate a year-round schedule for the more complex satellites.

When there is more than one decision to be made at decision points, the third input to the planner, the control knowledge (declaratively expressed as control rules) guides the problem solver to the correct branch of the search tree avoiding backtracking.

Acknowledgements

We want to thank all the HISPASAT Engineering Team for their help and collaboration shown during the developing time of this project, especially Pedro Luis Molinero and Arseliano Vega. This work was partially funded by the CICYT project TAP1999-0535-C02-02.

References

[Borrajo, Vegas, and Veloso 2001]. Borrajo D., Vegas S. and Veloso M. *Quality-based Learning for Planning*. Working notes of the IJCAI'01 Workshop on Planning with Resources. IJCAI Press. Seattle, WA (USA). August 2001.

[Chien et al 2000] Chien S., Rabideau G., Knight R., Sherwood R., Engelhardt B., Mutz D., Estlin T., Smith B., Fisher F., Barrett T., Stebbins G. and Tran D. *ASPEN Automating Space Mission Operations using Automated Planning and Scheduling*. SpaceOps 2000, Toulouse, France, June 2000.

[Veloso et al 1995] Veloso M., Carbonell J., Perez A., Borrajo D., Fink E., and Blythe J. *Integrating planning and learning: The PRODIGY architecture*. Journal of Experimental and Theoretical AI, Vol 7, pp 81-120, 1995.