N94-23967

PLANNING: MANAGEMENT OF PREDICTABILITY AND UNCERTAINTY AND KEEPING ABREAST OF DEVELOPMENTS

Christophe BASTIEN-THIRY

Centre National
d'Etudes Spatiales

18, Avenue Edouard Belin
31055 Toulouse Cedex, France
Tél.: 61.27.38.53 - Fax: 61.28.26.03

Gérard VERFAILLIE

Centre d'Etudes et
de Recherches de Toulouse

2, Avenue Edouard Belin
31055 Toulouse Cedex, France
Tél.: 61.55.71.95 - Fax: 61.55.71.32

ABSTRACT

The purpose of this study is to propose method to set up and control of a space mission plan such as that of the HERMES spaceplane. The interest of this subject, other than its complexity, is due to the need to manage imprecision and uncertainty during a mission, as well as changes in between missions. Under these conditions, the set up and control of a flight plan require certain special attention and this has led us to define a certain number of qualities:
* Mastery of complexity in order to resolve conflicts between activities : configuration, resource and time management,
* Consideration of various criteria such as risk minimization or the attainment of mission objectives.
* Robustness and flexibility to allow for hazards and deviations from the norm during operation without having to draw up new plans.
* Aptness for replanning by making changes to the plan without having to set up the whole plan again.
* Memorization and explanation facility in order to manage developments between missions.

Key Words: scheduling, robustness, flexibility, revision, explanation, design process.

1. MOTIVATION

The study for which we present certain results here is being carried out mainly by the CERT under CNES's Research and Technology activity. As the study is not yet finished, it will be first results that are presented in this article.

Planning and scheduling are traditional, reputedly difficult subjects of study which concern a much wider community than simply those involved in the space field. Moreover, many tools and methods have already been put forward as an effective response to some of these problems. So two questions come to mind: what specific constraints are imposed by our space activity, and what justifies the need for progress in this domain?

Space activity, as we know, is expensive and uses ever more complicated technological means. This obliges all project designers to consider not only the feasibility and safety aspects but also the optimization of resources as an absolute necessity. What is more, the urgency of this necessity leads to higher order reflection. Tomorrow's requirement will no longer be "to design something technically and economically feasible" (with optimization being taken into consideration afterwards) but to "think of another way designing things so as to master complex systems (in technical, human and economic terms)". In this perspective, the optimization aspects must be considered as a point of view intervening from the very start of the design process in the same way as techniques and safety.

Our thoughts and the studies performed on planning and scheduling tools and methods suggest elements of a solution to this problem. We would be justified in saying that not only do these tools obviously enable a satellite's resources to be optimized after the event but that the methods also assist the design process for space systems. Indeed, the planning and scheduling methods (and also, for example, the diagnostic tools and methods) should influence design choices: how can a system be designed without considering the means of exploiting and controlling it?

2. PROBLEMS POSED IN THE FRAMEWORK OF A SPACE MISSION PLAN

The aim of the work we present falls within the perspectives mentioned above. However, from a pragmatic point of view, we preferred to start with a concrete study corresponding to an actual need. During the preliminary definitions of the study in 1990, the application that appeared the most propitious support for this research was the setting up and control of the flight plan for a Hermes mission.
Although the study is influenced by the problem of the Hermes flight plan, the study is being carried out with the objective of making it generic and reusable for other space missions, and, as we have seen, in the perspective of the general set of problems concerning the design process.

The management of activities during a space shuttle flight is fully planned in advance and updated as the flight progresses (Ref.1). The planning of the Hermes mission (definition of the activities to be performed to attain mission objectives) appeared to fall mainly within the scope of human thought rather than being suitable for automation.
Once the activities capable of meeting the mission objectives have been defined, the problem is to schedule them taking account of constraints connected with time and resource utilization: constraints for performing the activities as a whole, constraints with use of systems, time constraints on and between activities and events, constraints connected with the cyclic nature of certain activities, constraints on the use of consumable or renewable resources. The activity model used for the scheduling contains imprecisions (duration and quantity of resources actually needed for the activities) and uncertainties (possibility of failures or need for new activities). So to avoid having to revise the plan too often at scheduling level, we propose a double approach:produce a **robust** plan and execute it with **flexibility**.

The problem is **dynamic**:
- when the flight plan is being set up, the user may want to add or remove certain constraints to improve a given aspect;
- during the flight, because of the drifts and hazards not totaly absorbed by the robustness or flexibility of the plan;
- between two flights because of changes in the system and mission, which must result in corresponding changes in the plan.

The dynamicity to be efficient involve revision, i.e find a new solution plan from the former one. From an operational point of view, it is easy to see that the aim is to limit the alterations to be made to the plan as far as possible, whence the need for a plan revision mechanism that can work on local modifications.

Finally, the plan resolution system must offer means of **explanation** so that those working on the mission in the plan elaboration or control phases have ways of interacting with the solution-seeking mechanism. They must know the origin of a situation, the consequences of a choice or a constraint and be able to reconsider choices or constraints.

3. FOR UNPREDICTABILITY AND UNCERTAINTY: ROBUSTNESS AND FLEXIBILITY

Since it is impossible to set up a global model of the world, it will always remain drifts and hazards during the flight execution; so, to reduce their effects, the chosen model must foreseen them. It is why we have introduced the complementary concepts of the robust plan and the flexible plan in order to limit the scheduling revision (Ref.2).

3.1 Robust plan

We consider here only a few of the possible definitions of a robust plan, each of which is connected with a different operational concept. A robust plan scheduling can be defined as a scheduling that reduces to a minimum:
- the maximum loss of attainment of objectives. This definition favours the stability of the objective attainment level.
- the probability of loss of attainment of objectives.
- the maximum change that it can be necessary to make to the scheduling to keep the same level of satisfaction of objectives. This definition favours the stability of scheduling.

These definitions bring in the notion of attainment of objectives, which can be defined from the degree of importance attached to each objective. The last brings in the idea of measure of change of a scheduling, an idea that must be given a sense from an operational standpoint. It is also necessary to have access to information relative to the probability that foreseeable events will occur.

3.2 Flexible plan

Unlike the previous approach, which makes choices but tries to make them as resistant as possible to the perturbations that will inevitably occur, the approach that we designate as flexible limits the choices to the parts of the scheduling that can be based on sufficiently precise and certain models of activities. For the parts where the models are imprecise or uncertain, the choices (like the precise start dates of activities) are delayed. These choices will be made either at the instant when the start of execution has sufficiently restricted the areas of imprecision and uncertainty or, failing this, at the time when the execution forces the choice to be made.

It must be said that these definitions and approaches that attenuate the problem of revising a scheduling are independent of the mechanism chosen for finding the solution and should rather be considered at the time when the mission, its objectives and the corresponding actions are designed, i.e. when the planning is done. In other words, it is the modelling of the problem that will ensure robustness and flexibility. It is worth noting in passing that the identification of the principles of a robust, flexible plan, which took place while the principles of a Hermes mission were being drawn up, allowed the concept of vehicle/ground system independence to be clarified (distribution of responsibilities between the vehicle and the ground segment).

4. FOR DYNAMICITY, EXPLANATIONS AND INTERACTIONS: REVISION, LOCAL MODIFICATIONS AND MEMORIZATION

Let us define the problem of scheduling as follows: Consider a set of tasks coming from the planning activity. The problem consists of scheduling them in time and allocating resources to them. These resources may have specific characteristics expressed as constraints on their use: they may be sharable, consumable, renewable. The tasks are generally linked to one another by linear constraints on the start and finish dates for their execution. Finally, some criteria can be expressed with respect to time (finish a task as early as possible) or to the use of resources (use the resource the least called upon).

There are several principles that can be applied to solve the scheduling problem thus defined. For the case where we are only interested in scheduling with respect to time constraints (no resource allocation: no disjunction), the best known principles are linear programming (Simplex algorithm) and the PERT method (search for the shortest path). In the cases where resource allocation as well as time must be taken into account, the methods envisaged are arborescent research (Backtrack, Branch and Bound algorithm), expert systems approaches (use of heuristics), constraint satisfaction, or random searching.

Before presenting the two solutions produced in this study in order to consider the dynamic and explicative aspects, we shall describe the principles of constraint satisfaction since it is on these principles that both are based.

4.1 Constraint Satisfaction

Our choice was that of **constraint satisfaction,** based on CSP (Constraint Satisfaction Problem) formalism (Refs. 2-3-4-5). What were the reasons for this choice? We recall that the aim of our studies was not so much to put forward new principles for solving the scheduling problem but rather to develop techniques enabling the solution to be found and, above all, offering the capacity to manage the dynamic aspects of the plan and propose possibilities of interaction with the user. Our problem was thus to choose a solution-seeking principle that was as efficient as possible and capable of being enriched with the target functions. Constraint satisfaction is a subject that has come under study fairly recently (latter half of the '80s) and, all in all, has been little explored. This type of programming quickly proved itself to be extremely efficient in solving highly combinatory problems, which is the main characteristic of scheduling problems. This technique thus offered assured efficiency and a potentially fertile field to be opened up in our sphere of research. Finally, numerous commercially available tools implementing the CSP method have come into being, thus reducing the development effort needed for our work.
CSP formalism can be presented as an extremely general form of problem solving, on the basis of which equally general methods of reasoning have been developed.
A CSP is defined by (Ref.6):
- a finite set V of variables v and

- a finite set C of constraints c.
- a finite domain D of possible values is associated with each variable v.

With each constraint c are associated:
- a subset V' of variables of V connected by the constraint and
- a relation R of any kind that defines the constraint to be respected between the variables of V'.

The basic problem of a CSP consists of finding, for each variable of V, a value such that all the constraints of C are respected. But other problems may also be set on the basis of this formalism:
- knowing whether or not a solution to the previous problem exists (without necessarily producing one),
- determining all the solutions or their number,
- producing a solution having a specific quality (maximizing a criterion, for example).

All these problems are known to be NP-complete, i.e. there is no algorithm of polynomial complexity that solves them. Their complexity increases exponentially with the size of the problem (number of variables, of constraints and their nature, cardinal of the domains associated with the variable). In spite of this NP-complete character, general methods propose interesting solutions. We present below the two most currently used methods.

4.1.1 Enumeration

The basic algorithm of **enumeration** methods (also called Backtrack method) is a simple arborescent research algorithm consisting of:
- determining an order of variables to be assigned,
- assigning the variables one after another, in the previously chosen order, choosing a value in its domain for each one,
- verifying after each assignment that the situation is consistent, i.e. whether the constraints linking the variable that has just been assigned to those previously assigned are respected,

until:
- all the variables have been assigned respecting the constraints or
- an inconsistency is observed, in which case a new value is tried for the current variable; if no further value is possible for this variable, we backtrack to the previous assignment. Of course, this unrefined algorithm does not avoid the combinatorial aspect of the problem and numerous improvements have been made, like the use of heuristics either for the choice of the order of assignment of the variables (e.g. choose the variables intervening in the greatest number of constraints first) or for the choice of the value to be assigned (Ref.7). Similarly, instead of backtracking to the previous assignment in case of failure, it is possible to avoid going back over assignments that are not involved in the inconsistency observed (Ref.8).

4.1.2 Filtering

The **filtering** method (also called constraint propagation or consistency enforcement (Ref.9)) uses the principle that any CSP can be transformed into an equivalent CSP that is simpler, even easy (a CSP is said to be easy if it can be solved by the Backtrack algorithm without actually backtracking). Two CSP's are said to be equivalent if any solution of one is also a solution of the other. One CSP is simpler than another if the domains of values associated with the variables and the relations associated with the constraints of the first are included in the domains and relations of the second.In practice, the consistency enforcement method is limited to **arc-consistency**. This method relies on the application of a simple procedure between two variables v_i, v_j which suppresses from the domain of v_i any value that makes it impossible to satisfy the binary constraints between v_i and v_j. This procedure is applied to all pairs of variables until no domain of variables remains to be reduced. It can be generalized to n-uplets but at a cost in computing time which is out of all proportion with the expected gain.

However, this method does not guarantee the existence of a solution and does not provide the means of producing one. In consequence, a reasonable method of obtaining a solution consists of alternating enumeration and filtering by arc-consistency (Ref.10).

4.2 First solution - improvement of the "enumeration plus filtering" method.

The principle of this solution is based on an alternance of enumeration (possibly caused by choices made by the user) and filtering. This principle is accompanied by the memorization of information on the origins of the removal of values. The method starts from the following observation: any removal of a value val_i from the domain of a variable v_i has two possible origins:
- the enumeration mechanism, i.e. a choice that excludes this value, or

- the filtering mechanism, i.e. there are no more values in the domain of a variable v_j, linked to v_i by a constraint c_{ij}, that support it.
The absence of support of val_i by a value val_j of the domain of a variable v_j linked to v_i by a constraint c_{ij} also has two possible origins:
- non-respect of constraint c_{ij} by the pair (val_i, val_j) or
- the removal of val_j from the domain of v_j.

We therefore suggest that, whenever a value val_i is removed from the domain of a variable v_i, the origin of the removal, enumeration or filtering, should be stored in memory. For the case of filtering, we shall store the origin of the absence of support for this value by another variable v_j, linked to v_i by a constraint. Thus, by recursion, it is possible to find the set of causes for the removal of a value or set of values, which allows us to find the origins of a situation. Similarly, to reconsider choices or constraints that led to a failure, it is of no use to pay attention to the choices or constraints that are not part of the set of causes identified as being responsible for the situation.

It should nevertheless be noted that the explanation we shall be able to give for the removal of a value will only be the first as, once a value has been removed, the mechanisms for solving CSP's do not pay attention to other possible ways of removing it.

This method thus enables explanations to be given. Furthermore it offers the possibility of managing dynamics. In the case of removal of a constraint or a choice, it is possible to put back at present all, and only all, the values of the variables such as this constraint or choice is a cause of their removal. A new attempt at filtering all these values can then be made in order to restart the solving process from a coherent state.

4.3 Second solution - improvement of the enumeration method: a local modification approach

The principle of this method is: when a failure is found with assignment a value to a variable, instead of reconsidering, as in the Backtrack algorithm, the last choice made for the variable previously assigned, which may have nothing to do with the failure, only those previous assignments that are incompatible with the present are reconsidered. For each of these disassigned variables, a new assignment is attempted with no possibility to modify the assignment of the variable that caused them to be changed. Success is obtained when all the disassigned variables have been able to be assigned again; if one of them cannot find a coherent assignment, the attempt has failed. Another assignments, if they exist, of the first variable then can be attempted. It can be shown that this method is complete (if a solution exists, it will be found).
Management of the dynamic aspect requires no additional measures:
- in cases where a constraint is removed:
  either the previous problem had a solution: it is necessarily a solution of the new, less constrained problem, or
  the previous problem had no solution (failure of the attempt to assign a variable): an attempt to solve the new problem may be started from the context (set of assignments) preceding the failure;
- in cases where a constraint is added:
  either the solution to the previous problem respects the new constraint and there is nothing to be done, or
  the solution to the previous problem does not respect the new constraint: it is then sufficient to choose one of the variables connected by the constraint, disassign it and attempt a new assignment.

The essential advantage of this method is that it enables a problem to be solved starting from any context and, in case of revision it finds a solution, if one exists, fairly close to the previous one since its principle is based on local modifications.

5. CONCLUSIONS AND PERSPECTIVES

Each of the methods presented answers a part of the problem. The first, although it offers a mechanism for revision and means of explanation, offers no way to revise localy. The second, although it offers an enumeration mechanism working indifferently in construction or revision mode and is able to find a new solution not to far from the former, has none of the filtering mechanisms which have so notably improved the efficiency of the Backtrack algorithm. Hence the search for a mixed method where the enumeration mechanism would be replaced by a mechanism based on local modifications to which could be added a filtering mechanism with storage of the causes of removal. Another way of mixing the methods is to apply them in different phases:

when the plan is being drawn up or adapted to a new mission, the first method is used, which allows interaction with the people involved; in the control phase where it is necessary to react fast and find a solution relatively close to the previous one, the second method is used, starting from a context established by the first method.

We would like to give a sense to the explanations these methods are capable of providing. By attaching to each constraint or task definition the reasons for defining them that way (e.g. the task "play back the recorders" was defined because there is no direct transmission during any part of the flight), we can give a set of justifications due to the context, the technology or any other consideration which tends to get forgotten as teams are renewed. Conversely, if modifications to the context appear over time, we shall be able to know the impact on solutions already found so as to be in a position to adapt and re-use them with full knowledge of the facts.

Finally, a design process being roughly an operation of finding a solution subject to constraints with re-use of the knowledge already acquired, we feel that it is possible to adapt and generalize the results of this study to the problem of designing complex systems.

6. REFERENCES

1. E.Bensana, G.Verfaillie and A.Gasquet.
Méthodologie d'élaboration et de contrôle de plan de vol pour navettes spatiales (rapport de fin de phase 1).
Technical Report 1/3393/DERI, CERT 1991.

2. E.Bensana and G.Verfaillie.
Méthodologie d'élaboration et de contrôle de plan de vol pour navettes spatiales (rapport intermédiaire de phase 2).
Technical Report 2/3393/DERI, CERT 1992.

3. T.Scheix, G.Verfaillie and B.Trousse.
Evaluation des outils de la programmation logique avec contraintes pour le traitement des problèmes à fortes combinatoire. (rapport de phase 1).
Technical Report 1/3411/DERI, CERT 1991.

4. T.Scheix, G.Verfaillie, B.Trousse and P.Charman.
Evaluation des outils de la programmation logique avec contraintes pour le traitement des problèmes à fortes combinatoire. (rapport de phase 2).
Technical Report 2/3411/DERI, CERT 1991.

5. T.Scheix, G.Verfaillie, B.Trousse and P.Berlandier.
Evaluation des outils de la programmation logique avec contraintes pour le traitement des problèmes à fortes combinatoire. (rapport de phase 2).
Technical Report 3/3411/DERI, CERT 1992.

6. U.Montanari.
Networks of constraints, fondamental properties and applications to picture processing.
Information Science, 7(2):95-132, 1974.

7. R.Dechter and J.Pearl.
Network-based heuristics for constraint satisfaction problems.
Artificial Intelligence, 34:1-38, 1988.

8. J.Gaschnig.
Performance measurement and analysis of certain search lalgorithms.
PhD thesis, Carnegie Mellon University, 1979.

9. A.Mackworth.
Consistency in networks of relations.
Artificial Intelligence, 8(1):99-118, 1977.

10. B.Nadel.
Constraints satisfaction algorithms.
Computer Intelligence, 5:188-224, 1989.