APGEN: A Multi-Mission Semi-Automated Planning Tool

Pierre F. Maldague
Adans Y. Ko
Dennis N. Page
Thomas W. Starbird

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove dr.
Pasadena, CA 91109
maldague@apgen.jpl.nasa.gov

Abstract

This paper discusses a multi-mission planning tool named APGEN, which is currently used by several Flight Projects at JPL. Although APGEN was not intended to function as an autonomous planner, it does meet stringent requirements imposed by current flight project customers. We discuss the nature of these requirements, how they are met in the current implementation of APGEN, and how we are planning to adapt APGEN to the closed-loop environment required by new interplanetary missions.

Introduction and Background

APGEN is a multi-mission planning tool currently in development at JPL. The requirements for APGEN were drawn from the experience of mission planners who were primarily interested in 'traditional' (non-autonomous) methods of commanding spacecraft. To establish the context in which the requirements for APGEN arose, let us start with a brief description of the key concepts of 'plan' and 'sequence'.

In a traditional commanding environment, the final product of the uplink system is a (time-ordered) sequence of time-tagged commands to be executed sequentially by the spacecraft. We refer to this as a 'sequence', and will not be concerned here with (important) details such as the precise representation of each command (ASCII vs. binary, packetized or not, etc.).

A plan, on the other hand, is a collection of activities, usually ordered according to increasing start times. An 'activity' can be defined loosely ('orbit insertion', 'imaging observation') or quite precisely (e. g. as a fully defined maneuver with 32 parameters specifying the timing of all related events, the delta V and change in attitude, etc.) Such a plan is summarized in a graphical timeline, each activity being a horizontal bar stretching from its start time to its end time.

JPL has developed a sequencing tool, SEQ_GEN, that allows uplink personnel to 'wrap' many commands into higher-level 'activities'. The spacecraft can be commanded at the higher, 'activity' level (which is much easier than trying to send the spacecraft individual commands). The process of 'expanding' activities into sequences is made automatic by SEQ_GEN. Without getting into details, the main steps in the 'adaptation' process that customizes SEQ_GEN to a specific project are as follows:

- defining the spacecraft commands
- defining activities and how they expand into commands
- defining the spacecraft model and how commands affect it
- defining the mission rules and implementing them as constraints

SEQ_GEN is currently being used by a variety of missions. From a planning perspective, however, SEQ_GEN is not an ideal tool, because not much can be done until all the adaptation steps have been carried out. SEQ_GEN is best when used for computation and analysis of details at the spacecraft command level; the SEQ_GEN user should really be familiar with the expansion rules as well as the nature of the spacecraft model in order to use it effectively.

Planning personnel felt that there ought to be a tool that allows them to produce 'rough' mission plans well before the details of the spacecraft commands are known. A key requirement is that this tool should *not* require the complex machinery of SEQ_GEN expansion and modeling before it could produce some meaningful results.

This tool was named APGEN, and the main requirements that it should satisfy are as follows:

- be easy and intuitive to use; in particular, easier to use than pre-existing sequencing tools such as SEQ_GEN and Plan-IT-II
- be able to represent simple resources such as Power and Fuel
- be able to model the effect of activities on resources in a simplified way
- be intuitive to operate
- be able to interface with sequencing tools, in particular SEQ_GEN (details were left unspecified)
- be able to operate in networked fashion, with several users sharing data

Interestingly, at the time APGEN was started, interviews with mission planners showed little interest in spacecraft autonomy, or even in autonomous planning on the ground. The primary goal was to provide mission planners (for the Cassini mission in particular) with a tool that could help a human planner well before the sequencing tools were 'adapted'.

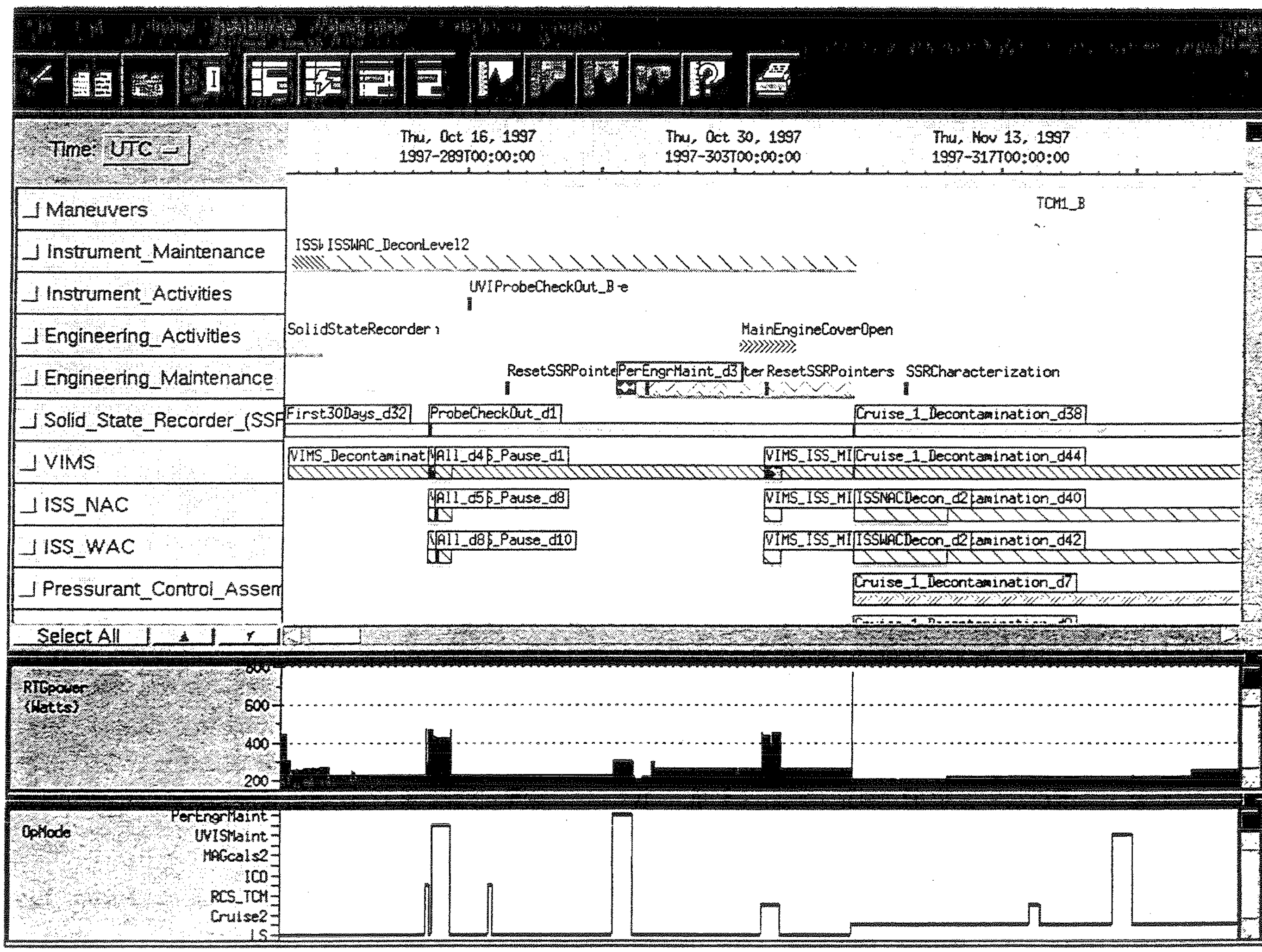

Figure 1: Partial View of the Cassini Inner Cruise Activity Plan as Displayed by APGEN

The Early Days: the Cassini Cruise Plan

APGEN was developed in a succession of engineering versions. As soon as APGEN became able to let an activity 'consume' a resource, Cassini planners started using it to put the Inner Cruise Activity Plan (IACP) together. The ICAP covers the part of the Mission Plan that extends from launch (October, 1997) until the switch to the High-Gain Antenna (February, 2000). The primary target audience for the ICAP is the Mission Planning Virtual Team (MPVT), which consists of 5 Systems Engineers plus representatives from all Science and Engineering Subsystems, or about 100 people in all. The Systems Engineers are in charge of both the adaptation (specifying resources, activity types and how they interact) and the actual plan. Plans are then communicated to remote sites (including several sites in Europe) where APGEN is used to view the plan. See Fig. 1 for a representative display of this plan.

The ICAP plan consists of about 250 activity types; these activity types contain 'usage clauses' that specify how they affect 40 different spacecraft resources. Although APGEN pretty much lived up to the expectations of its users, there were interesting surprises in the particular way the Cassini planners decided to use APGEN capabilities.

The first such surprise has to do with decomposition, which lets users switch between high-level and low-level views of (usually complex) activities. The designers of APGEN envisioned that adapters would use this feature for implementing successive versions of the same activity, each one more refined than the previous one. For example, a crude activity entitled 'orbit insertion', with a simple specification of its average resource use, might later be decomposed into a more detailed 'maneuver' with more realistic parameters, and with a more realistic pattern of resource usage.

In reality, however, Cassini planners ended up using decomposition to define hierarchical links between activities that influence resources simultaneously (a relationship named 'non-exclusive decomposition' in the APGEN jargon). This allows users to represent a complex maneuver as a single activity bar in the timeline, and yet to see the actual, detailed influence of that maneuver on all the resources it impacts as dictated by the sub-activities it contains.

A second surprise is that only about half the resources defined in the ICAP are physical resources that can be used to express planning constraints. The other half is intended to convey information about the plan. For instance, a scientist interested in a specific instrument can focus on the periods of activity of his/her instrument by looking at a resource that specifies the power mode for that instrument ('on', 'off', 'standby' etc.).

Perhaps the biggest surprise for the Cassini team was to realize that their APGEN plans would not automatically convert into a sequence acceptable to their sequencing engine (SEQ_GEN). APGEN has an option for producing files containing 'activity requests' in the proper format for input into SEQ_GEN. However, these files are meaningless unless they match the SEQ_GEN 'adaptation files', which specify the types of parameters of each activity as well as how each activity expands into subactivities and commands. Only after the first few request files were generated by APGEN did it become obvious that coordination between the planning and sequencing teams was essential to obtaining a smooth interface.

New Customers: the Push for More Planning Automation

New potential customers for APGEN include the Space Infra-Red Telescope Facility (SIRTF) and X2000. Both projects are seeking cheaper, more automated ways to conduct space missions. This presents APGEN with two new challenges:

- automatic sequencing of large numbers (thousands) of requests for observations
- modeling of activities whose expansion depends on the state of the spacecraft (conditional expansion)
- more generally, support closed-loop operation as opposed to the open-loop environment typical of ground operations

To meet the first challenge, new features for task scheduling have been incorporated into APGEN. In a nutshell, this allows APGEN to schedule certain activities only when certain conditions are met for a specified length of time. This capability does not make APGEN into a full-fledged 'automatic planner' (it still lacks the capability to refine and optimize proposed schedules). However this incremental approach has the merit of being quite fast and deterministic/predictable. Fig. 2 below shows an eample of this scheduling capability applied to the SIRTF project.

There is one area in which APGEN's 'traditional personality' provides it with an advantage over more modern planning tools: processing speed. Prompted by Cassini users who wanted quick access to their two-year Cruise Plan, APGEN developers had to introduce a number of time-saving features into the design. For example, the original design of

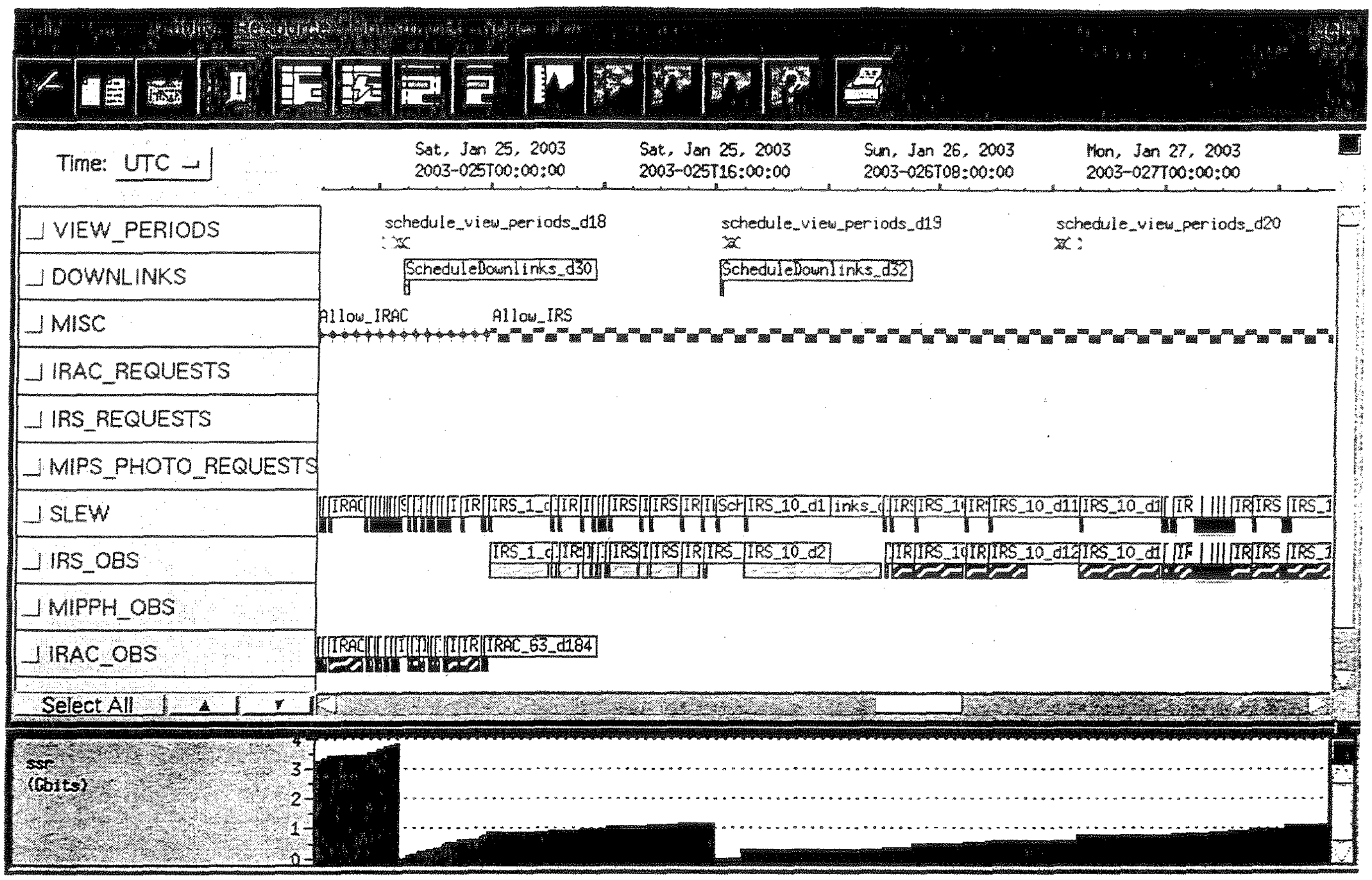

Figure 2: Partial View of a Preliminary Activity Plan as Scheduled by APGEN

APGEN featured an interpreter which was invoked every time the adaptation data had to be consulted. In the new design, the parser saves the parsing trees associated with adaptation data in the form of 'pseudo-code'. When adaptation data need to be consulted, APGEN 'executes' the pseudo-code without having to re-interpret the original data. A second example is provided by the many lists used to store activity types, activity instances, resources, constraints and many of their class members. The original design used a simple linked list scheme, which was adequate for testing purposes but brought the program to its knees when dealing with large numbers of objects. In the current design, the linked lists have been replaced by much more efficient balanced trees. The AVL algorithm (Knuth 1973) is used to insert and delete elements in logarithmic time (i. e., execution time grows as log(N) where N is the size of the list). These two changes resulted in a hundredfold performance improvement.

The second and third challenges are more difficult, because they amount to replacing fixed, time-tagged commands by a 'program' containing conditional (and perhaps iterative) statements. How does one validate such a beast? We don't know.

The Future: ASPEN, SEQ_talk, Autonomy

There are three areas of development planned for the next year. One is to enable APGEN to execute SEQ_GEN and access its results. In this way, as a project's development proceeds to the point of having adapted SEQ_GEN with its detailed models, results of such detailed modeling can be used in APGEN's planning. APGEN would serve as the principal interface to the user, but computations would be shared by the two programs. The programs will be linked by socket communication, leaving them loosely coupled, able to proceed independently with their own developments of other features. Similar linkage, referred to as SEQ_talk, to other existing sequencing and planning tools is being considered. The ability for APGEN to talk to other programs would help to solve one of the surprises encountered by the

Cassini project; computation would be done by the program best suited, but the results would be available to APGEN and its user.

Another area of development is to add access by APGEN to more intelligent planning capability. Use will be made of ASPEN (Automated Scheduling and Planning Engine), which includes a library of automated planning functions, and which is under development at JPL. Again, a loose coupling mechanism based on sockets will be used, enabling parallel development of APGEN and ASPEN.

The third area of development concerns the migration path from ground to flight software. We have recently proposed a concept which we call 'Just-In-Time Planning', in which an APGEN-like planner running on board the spacecraft would handle short-term scheduling tasks that require fast turn-around, such as faults and real-time events. This short-term planner would operate under the direction of a resource allocation manager, which could eventually be extended to a separate, on-board long-term planner.

Acknowledgments

Many of the concepts implemented in APGEN were prototyped in a program called Plan-IT-II, implemented at JPL. Plan-IT-II has a long history of pathfinding experimentation in concepts useful for practical automated planning. APGEN is funded by the Telecommunications and Mission Operations Directorate at the Jet Propulsion Laboratory, operated by the California Institute of Technology under contract by the National Aeronautics and Space Administration.

References

Knuth, Donald E. 1973. *The Art of Computer Programming, Volume 3: Sorting and Searching*. Reading, Mass.: Addison-Wesley.