N93-18677

Planning, Scheduling, and Control for Automatic Telescopes

Mark Drummond, Sterling Software
Keith Swanson, NASA
John Bresina, Sterling Software
Andy Philips, Sterling Software
Rich Levinson, Recom Software

NASA Ames Research Center
Mail Stop: 269-2
Moffett Field, CA 94035

1 Introduction

Making observations through telescopes is an activity of central importance to NASA. Whether a telescope is located on the Earth, is in orbit around the Earth as a satellite, is located on the moon, or is even on another planet, it presents an exciting and sometimes unique opportunity for gathering data about various astronomical phenomena. Telescopes have always been a scare resource, and astronomers have had to make do with extremely limited access. Further, an astronomer has been expected to be physically present at a telescope in order to gather data. Restricted access and local operation have limited the amount of data that can be gathered, and thus have directly contributed to fewer scientific results than might otherwise be expected.

Recent work by the Fairborn Observatory and AutoScope Corporation has freed astronomers from the need to be physically present at the telescope site. These organizations, working with astronomers, have designed and built control systems and associated hardware for the management and control of photoelectric telescopes; for a review of these Automatic Photoelectric Telescopes, or APTs, see Genet and Hayes (1989). While existing automation deals primarily with photoelectric telescopes, other sorts of telescope and other sorts of science are currently under investigation. The key point is that there is a perceived need, *within the astronomy community*, that the automation of local telescope control is desirable. Existing automation does not address all needs of all astronomers, but it does provide an excellent starting point. The eventual goal is what we call a "simplified management structure". The term refers to an approach to the management and control of telescopes that minimizes the number of people that must come between an astronomer's scientific goals and the telescopes required to realize those goals. A simplified management structure requires significantly more sophisticated telescope automation than is currently possible.

The Entropy Reduction Engine (ERE) project, carried out at the Ames Research Center, is focusing on the construction of integrated planning and scheduling systems. Specifically, the project is studying the problem of integrating planning and scheduling in the context of closed-loop plan use. The results of this research are particularly relevant when there is some element of dynamism in the environment, and thus some chance that a previously formed plan will fail. After a preliminary study of the APT management and control problem, we feel that it presents an excellent opportunity to demonstrate some of the ERE project's technical results. Of course, the alignment between technology and problem is not perfect, so planning and scheduling for APTs presents some new and difficult challenges as well.

This paper presents an argument for the appropriateness of ERE technology to the planning, scheduling, and control components of APT management. The paper is organized as follows. In the next section, we give a brief summary of the planning and scheduling requirements for APTs. Following this, in section 3, we give an ERE project precis, couched primarily in terms of project objectives. Section 4 gives a sketch of the match-up between problem and technology, and section 5 outlines where we want to go with this work.

2 APT problem summary

An Automatic Photoelectric Telescope is a telescope controlled by a dedicated computer for the purpose of gathering photometric data about various objects in the sky. While there are many sorts of photometric techniques, we focus on the technique known as *aperture photometry*. An excellent overview of aperture photometry is given by Hall and Genet (1988). In aperture photometry, and for current purposes, a *group* is the primitive unit to be scheduled. A group is a sequence of telescope and photometer commands defined by an astronomer. Any given astronomer has certain scientific goals, and he or she uses the group as the primary unit of instruction to an APT in order to achieve those goals. The language used to define groups is called ATIS (for Automatic Telescope Instruction Set); ATIS is an ASCII-based language for communicating with APTs (the *de facto* standard).

The communication process between astronomer and APT proceeds roughly as follows. First, an astronomer who wishes to use an APT forms a set of groups consistent with his or her scientific goals. These groups are written specifically in terms of a given telescope: since each telescope can vary slightly (instruments, optical characteristics, mechanical characteristics, location on the Earth), groups must be formulated in a telescope-specific manner. For any given APT there is a single person who acts as a central clearing-house for usage requests; such a person is known in the vernacular as the APT's *Principal Astronomer*, or PA. Thus, once an astronomer has assembled his or her set of ATIS groups, they package the groups off to the appropriate PA. The PA collects together such sets from a variety of astronomers, attempts to ensure that the telescope is not overloaded, and then sends the complete set of groups off to the correct telescope. Actual communication between PA and APT is carried out by using personal computers, modems, and phone lines, but the particular technology isn't critical for the current discussion. The important aspect of the communication is that the PA can be located anywhere on the planet (in principle), and need only have access to an appropriate communication link.

The PA sends a set of groups to an APT, with the intention that these groups should be run for some time; eventually, the PA requests from the telescope the results that have been obtained under the execution of the given groups. The elapsed time varies, and depends on the telescope, the groups, the PA, and a variety of other factors. Of course the goal is to worry the astronomers (and the PA) as little as possible about the picayune details of day-to-day telescope management. Thus, the telescope is often left alone for significant periods of time (weeks, perhaps months). However long the telescope operates unattended, it is eventually asked for data, and this is returned to the PA as a "results file". The results file is also in the ATIS language, and it contains the groups that were executed, relevant observing parameters to help with data reduction, and the actual data obtained from the observations. The PA breaks this results file into the pieces that are relevant for the astronomers and sends each astronomer the results of his or her requested observations. Thus the cycle of group submission, compilation, execution, and data return can begin again when the astronomers discover that the data they've been given doesn't really tell them what they wanted to know (such are the joys of real science).

Of course, the interesting part of this process is the part that we've completely ignored so far; that is, the process by which the groups are accepted and executed by the local telescope controller. *This* is the interesting part, and it is with respect to this process that our planning and scheduling work can make a real difference. Currently, a program called ATIScope manages the execution of a file of groups. ATIScope runs locally at the given telescope, using observatory and telescope sensors to determine when to execute the provided groups. ATIScope has a variety of responsibilities, but we focus specifically on only one of these; namely, *group selection*.

At the core of ATIScope is a test that attempts to find a "currently" executable group. Roughly, a group is executable if the logical preconditions established by its astronomer-creator are met. Typically, these preconditions relate to the current date and time and to whether the moon is up or down. Additionally, an astronomer can specify a group *priority*, used by ATIScope to sort the groups in order of importance. There are other pseudo-preconditions that have to do with frequency of group execution, but we can safely ignore these for now.[1] Roughly, the core of ATIScope is a sense-check-execute loop. In sensing, all relevant environmental parameters are determined (date, time, moon status). ATIScope next checks to see which of the various possible groups are enabled according to the match between the current sensor values and the astronomer-provided preconditions. Let's call the set of groups that pass this matching test the *enabled* groups. The set of enabled groups is winnowed by the application of *group selection rules*. These rules express heuristic knowledge relating to the wisdom of executing any particular group before any other. In scheduling parlance, this scheme is sometimes called *heuristic dispatch*, since at any point in time, some task (here, a group) is "dispatched" for execution, and the selection of a task is determined, purely locally, by the application of some domain-specific heuristics. The information content of the heuristics used by ATIScope isn't critical for the current discussion (however, see Genet & Hayes, 1989, pp. 207–210). In the current context, heuristic dispatch is used to transform the set of enabled groups into a (hopefully) single group that is executed. If the heuristic group selection rules fail to winnow the set of enabled groups down to a single candidate, then the first group in the given list is selected (this, however, almost never happens, as the group selection rules normally produce a single preferred group). Following selection, the lucky group is executed, at which point telescope control is largely surrendered to the astronomer who wrote the group. Of course, there are safety checks to ensure that the astronomer's commands don't damage equipment, but if the commands are well-behaved (and if the weather cooperates), group execution finishes normally, and ATIScope is free to perform another iteration through its sense-check-execute loop.

How well does ATIScope do, in terms of schedule quality, by using this heuristic dispatch technique? One way of answering this question is to recall the old adage about an incredible dancing dog: the question of the quality of the dog's dancing needn't really be raised; one should instead be happy that the dog dances at all. ATIScope does, of course, provide an acceptable level of performance for some astronomers. There is no question, however, that the level of telescope performance can be dramatically improved by better group scheduling. With the heuristic dispatch technique, all decisions are *local* in the sense that no temporal look-ahead is performed to evaluate the ramifications of executing a given group. The system also has no memory of what it has done on previous nights, so groups cannot be selected with respect to some desired frequency of execution. Other scheduling techniques, such as those based on *temporal projection* (Drummond & Bresina, 1990), consider the impact of a given action by looking ahead in time to see how the current local choice impacts global objectives. Look-ahead is only sensible when astronomer objectives can be clearly and precisely formulated. Assuming that this can be done, it seems clear that a look-ahead scheduler

[1] The main factors that influence frequency of execution are a group's *probability* and *number of observations*; see Genet & Hayes (1989), p. 208.

can outperform the current ATIScope heuristic dispatch method. ATIScope, however, provides us with an existing level of performance against which all would-be contenders can be gauged.

3 ERE goals

The design of systems that can synthesize plans has been a long standing research topic in the field of Artificial Intelligence (AI). Such systems, called *planners*, are given a description of the problem at hand, and can synthesize a plan to solve that problem. Of course, a plan is merely a specification of a solution, and so must be executed to *actually solve* the given problem. Various sorts of "execution system" are possible; for instance, a plan might be executed by a manufacturing system, by a group of people, or by a robotic device; all that is required is a system that is capable of instantiating the plan's actions and thus producing the desired result. The design of these automatic planners has been addressed in AI since its earliest days, and a large number of techniques have been introduced in progressively more ambitious systems over many years. In the AI research branch at NASA Ames, the Entropy Reduction Engine (ERE) project is our focus for extending these classical techniques in a variety of ways. In this section we present the ERE project's overall goals; for more detail on the architecture itself, see Bresina & Drummond (1990), Drummond & Bresina (1990a, 1990b), and Drummond, Bresina, and Kedar (1991).

The Entropy Reduction Engine project is a focus for research on planning and scheduling in the context of closed-loop plan execution. The eventual goal of the ERE project is a set of software tools for designing and deploying integrated planning and scheduling systems that are able to effectively control their environments. To produce such software tools, we are working towards a better theoretical understanding of planning and scheduling in terms of closed-loop plan execution. Our overall project has two important sub-goals: first, we are working to integrate *planning* and *scheduling*; second, we are studying plan execution as a problem of *discrete event control.* Let's consider these complementary goals in a bit more detail.

Integrate planning and scheduling. Traditional AI planning deals with the selection of actions that are relevant to achieving given goals. Various disciplines, principally Operations Research, and more recently AI, have been concerned with the scheduling of actions; that is, with sequencing actions in terms of metric time and metric resource constraints. Unfortunately, most of the work in scheduling remains theoretically and practically disconnected from planning. Consider: a scheduling system is given a set of actions and returns, if possible, a schedule composed of those actions in some specific order. If the scheduler cannot find a satisfactory schedule, then it simply fails. The business of planning is to *select* actions that can solve a given problem, so what we need is an integrated planning and scheduling system to overcome the problems of scheduling alone. An integrated planning and scheduling system would be able to consider *alternative* sets of actions, unlike the stand-alone scheduler, which is unable to deviate from its given action set. We are working towards such an integrated system by incrementally constructing a unified theory of planning and scheduling that can be computationally expressed as practical software tools.

Study plan execution as a control theory problem. Most planning and scheduling work assumes that the job of the automatic system is done when a plan or schedule has been generated. Of course, one of the first things that you learn about plans is that they are rarely ever perfectly predictive of what will happen. As Dwight D. Eisenhower observed, "Plans are nothing, planning is everything". We agree with this view, since it tells us that the importance of planning does not lie in the existence of a *single* plan, but rather in a system's ability to re-plan and predictively manage plan execution failures in light of feedback from the environment. In the ERE project, we view plan execution as a problem in discrete event control; specifically, we formalize a plan as a simple type of feedback controller, and this gives us a new view on plan execution. Traditionally, plans have been executed by executing each component action in sequence. Our plans are functions that map from current sensor values and a desired goal into a set of acceptable control actions. The interpretation of the function is that any of the actions, if executed in the current situation, constitute an acceptable prefix to a sequence of actions that eventually satisfies the goal.

4 The match, in the abstract

The previous two sections have, in rough terms, explained the APT problem and overall ERE project goals. In this section, we consider how ERE technology promises to address key APT planning and scheduling issues. This section is optimistic and is, by necessity, "promissory", in the sense that some of what we suggest has yet to be rigorously demonstrated. This section reflects what we currently perceive as opportunities for using ERE technology on the APT planning, scheduling, and control problem.

First, the obvious: ERE is an architecture for producing systems that look ahead into the future, and by so doing, choose actions to perform. We feel that the ERE architecture is well-suited to the APT planning and scheduling problem in this regard. ATIScope currently does no look-ahead, so assuming that our system does, it should be able to produce better schedules. In fact, one of our research interests is the relationship between the cost of looking ahead and the increased "quality" of the system's actual behavior. In the APT domain, the quality of system behavior is determined by the amount and quality of the data returned by a given set of observations, and by the fairness of telescope allocation to the various astronomers' groups. Now ATIScope currently achieves a particular level of quality, and we expect to be able to increase this through some amount of look-ahead. But at what cost? When does look-ahead actually give rise to better system performance? ATIScope, while perhaps not producing the highest quality behavior, does so with great alacrity. A scheduling system that does any amount of look-ahead consumes more computational resources than ATIScope, so the behaviors it produces had better be worth the increased cost. Of interest here is the impact of environmental factors on the underlying requirement for look-ahead: if the environment is completely predictable, and if a great deal of time is available in advance, then a scheduler that looks ahead extremely far into the future is apparently what's required. However, if the environment can change quickly, and change in unpredictable ways, then much of the work done by a look-ahead scheduler is wasted. The correct balance between look-ahead and heuristic dispatch is truly a function of the domain. There has been little empirical study of this issue in general, and we feel that APT planning and scheduling provides an excellent test case.

We have an algorithm for incremental, "anytime", planning (Drummond & Bresina, 1990) that we think will be useful in the APT context. While our algorithm has only been tested on relatively simple planning problems, we think that many of the underlying ideas transfer to scheduling as well. The essential idea is as follows: if a system has a limited amount of time to plan, and, having planned, is allowed to plan no further, then it makes sense for the system to make the best use of the available time by incrementally improving its current plan until time runs out. Our algorithm, called *traverse and robustify*, does this. It uses information about possible execution outcomes to predictively patch errors, before they actually occur. By doing this the algorithm attempts to maximize the probability that the plan it finds will satisfy the user's objectives. This algorithm promises to be useful in a scheduling context, and APTs provide an appropriate test-domain. If we think of the scheduler as running during the day (remote from the telescope, in the PA's place of work), and imagine that the finished schedule will be shipped to the telescope for overnight execution, then one would like the schedule produced to be of the highest possible robustness given the available time, so our algorithm seems appropriate.

5 Objectives

First and foremost, we must define an appropriate objective function for APT observation schedules. How well can this objective function be formalized? How will we notate it? That is, what will be our language for writing down the objective function? For the problems we have studied to date, our language of *behavioral constraints* has been adequate. The current behavioral constraint language allows a user to give arbitrary conjunctions and disjunctions of predicates that must be maintained true (or prevented from being true) throughout an interval of time (see Drummond & Bresina, 1990, for more detail). Is this language adequate for expressing the sorts of goals that astronomers have? Will we need to drop into the language of arbitrary mathematics? Of course, this *is* what most of decision analysis does, so should we expect to do any better? We hope to devise a new sort of behavioral constraint language, specifically designed to allow astronomers to define APT observation schedule preferences. Even with such a specially-designed language, there's a remaining second-order problem: the PA (or other user) must be able to define what constitutes a fair and equitable tradeoff of telescope and instrument allocation between different astronomers. Of course, we don't want a person (the PA or other user) to have to specify the *specific* tradeoff for each given scheduling instance, but the *general form* of the tradeoff function used must be defined by a user. These and other interesting issues lurk in the vicinity of schedule objective functions.

We are fortunate to have access to several APT experts. One expert is an original APT architect who has founded a firm to commercially produce APTs. The other experts are experienced photometric astronomers, one of whom is an active APT user and has acted as a principal astronomer in the past. It is our hope that by working directly with this diverse and experienced group of APT developers and users, we will be able to produce planning and scheduling tools of use to a large number of photometric scientists.

In the short term (6 months), we plan to produce an interactive scheduling tool for use by ourselves, with our APT user acting as a local domain expert. The tool will help a user analyze a given set of groups by interactively determining the best sequence in which the groups should be run, providing help with the selection of the best sequence, but leaving the user free to intervene should he or she so desire. The system will automatically compile out a set of group selection rules that will produce the desired set of group execution sequences. Essentially, our system will be used to compile a set of scheduling dispatch rules that are designed specifically for the target set of groups, to be run on the target telescope, for a particular night of observations. We have studied the problem in some detail and are confident that our existing techniques for compiling such rules will work on the APT problem (see Drummond, 1989).

We have access to an APT simulator and will use this to evaluate our system's evolving capabilities. Of course, the eventual goal of this research is to remove humans from the control loop, so this first short term objective might not appear to be a tremendous step forward. It is, in fact, best construed as a step "sideways", prefatory to a giant *leap* forward. We will use our interactive scheduling tool to gain experience with the APT planning and scheduling problem; our eventual goal is to entirely automate the decisions still made by a human user. This first sideways step towards a decision support system is thus not an end in itself, but only a means to a bigger, more important end.

In the medium term (1 year), we plan to produce a better, incremental scheduler designed to replace the ATIScope system. Our new scheduler would be based on experience gained with building our look-ahead scheduling decision-support system. Our scheduler, like ATIScope, would accept a set of groups from the PA (or various astronomers, thus freeing the PA entirely from any scheduling responsibilities), and would schedule and execute these in a flexible manner. This first prototype automatic scheduler would not provide a very sophisticated language of scientific objectives; instead, it would allow a user or users to specify a set of groups, and would attempt to better the current level of performance obtained by ATIScope by doing temporal projection (look-ahead) and history recording (remember-behind).

Our long term plan (2 years) is to extend the language of objectives to allow users to specify interesting scientific objective functions. The first test case would be a facility for filling out a desired light curve. Other test cases will be established in conjunction with our APT experts. The extra functionality offered at this stage of development will be that of *planning*, as opposed to pure *scheduling*. It is at this point that our system really begins to offer increased scientific power over that of the traditional ATIScope-style system. Until now, we have only sought to increase the "quality" of the group execution sequences. Here, we seek to increase the expressiveness of the language that is used by an astronomer to specify scientific objectives.

Once individual APTs are routinely being used by remotely located astronomers, with nearly all scheduling conflicts being resolved automatically, many new opportunities arise. For instance, at this point it becomes practical to consider a network of relatively inexpensive telescopes, located around the world, which are able to provide continuous observation of astronomical objects. While possible now for exceptional events (supernova), the logistical overhead precludes wider practice.

We are purchasing and intend to operate a 16-inch APT. This telescope will be located in northern California, and will be made available to members of the scientific community, with the focus being on educational institutions. We will make our system available over the InterNet, such that remotely located astronomers can simply Email request files to our system. Our system will accept a number of requests from various users, schedule them, and download the set of groups and group selection rules to the telescope. Users will receive their requested data via return Email or will be given access to an FTP site where their data may be recovered. This system will provide the first example of a totally automated telescope planning, scheduling, and control system. We plan to have the system operating totally autonomously as soon as possible.

We hope that our demonstration of fully automatic telescope operations will serve as groundwork for new applications of simplified telescope operations. Of particular interest is the possibility of placing a number of small telescopes on the moon (Genet *et al*, 1992). Such a telescope facility would be an excellent test of our "simplified management structure". We feel that ERE can provide a solid base for the development of integrated telescope planning, scheduling, and control systems that help to make this simplified management structure a reality.

Acknowledgements

Comments from Russ Genet, David Genet, Butler Hine, and Bill Borucki have been extremely useful; to each of them, our thanks.

References

[1] Bresina, J., Drummond, M., and Kedar, S. *Forthcoming.* Reactive, Integrated Systems Pose New Problems for Machine Learning. To appear in the volume on Learning in AI Planning and Scheduling Systems; Langley, P., and Minton, S. (eds).

[2] Bresina, J., and Drummond, M. 1990. Integrating Planning and Reaction: A Preliminary Report. *Proceedings of the AAAI Spring Symposium Series* (session on Planning in Uncertain, Unpredictable, or Changing Environments).

[3] Drummond, M. 1989. Situated Control Rules. *Proceedings of Conference on Principles of Knowledge Representation & Reasoning.* Toronto, Canada.

[4] Drummond, M., and Bresina, J. 1990a. Anytime Synthetic Projection: Maximizing the Probability of Goal Satisfaction. In proc. of *AAAI-90.*

[5] Drummond, M., and Bresina.J. 1990b. Planning for Control. In proc. of *Fifth IEEE International Symposium on Intelligent Control*, published by the IEEE Computer Society Press, Philadelphia, PA. pp 657–662.

[6] Drummond, M., Bresina, J., and Kedar, S. 1991. The Entropy Reduction Engine: Integrating Planning, Scheduling, and Control. *Proceedings of the AAAI Spring Symposium Series* (session on Integrated Intelligent Architectures).

[7] Hall, D.S., and Genet, R.M. 1988. Photoelectric Photometry of Variable Stars. Wilmann-Bell, PO Box 35025, Richmond, VA (2nd edition).

[8] Genet, R.M, Genet, D.R., Talent, D.L., Drummond, M., Hine, B., Boyd, L.J., and Trueblood, M. 1992. Multi-Use Lunar Telescopes. A chapter in "Robotic Observatories in the 1990's". Edited by Alexei V. Filippenko, published by the Astronomical Society of the Pacific Conference Series.

[9] Genet, R.M., and Genet, D.R. 1991. Dynamic Scheduling of Astronomical Observations By Intelligent Telescopes (An Informal Discussion of the Problem). Draft Paper, AutoScope Corporation, Mesa, AZ.

[10] Genet, R.M., and Hayes, D.S. 1989. Robotic Observatories: A Handbook of Remote–Access Personal–Computer Astronomy. Published by the AutoScope Corporation, Mesa, AZ.