

Design and Implementation of An Experiment Scheduling System for the ACTS Satellite

N95-23756

Mark J. Ringer

NYMA Inc.

NASA Lewis Research Center Group

Cleveland, Ohio 44142

(216) 433-8060

Ringer@mars.lerc.nasa.gov

KEY WORDS AND PHRASES

Planning and Scheduling

INTRODUCTION

The Advanced Communication Technology Satellite (ACTS) was launched on the 12th of September 1993 aboard STS-51. All events since that time have proceeded as planned with user operations commencing on December 6th, 1993. ACTS is a geosynchronous satellite designed to extend the state of the art in communication satellite design and is available to experimenters on a "time/bandwidth available" basis. The ACTS satellite requires the advance scheduling of experimental activities based upon a complex set of resource, state, and activity constraints in order to ensure smooth operations. This paper describes the software system developed to schedule experiments for ACTS.

DOMAIN DESCRIPTION

ACTS is a next generation communication satellite that incorporates three main technical gains: Demand Assigned Multiple Access - Time Division Multiple Access (DAMA-TDMA) with very small (0.3°) hopping spot beam antennas, use of Ka Band (30/20 GHz), and onboard processing. The DAMA-TDMA beam-hopping network allows multiple geographically distributed users to access the

satellite virtually simultaneously with smaller aperture antennae. On-board processing allows rain-fade alleviation algorithms to be added to the communication path since the Ka band is more susceptible to attenuation by rain. Very high data rates are possible in the Ka band, these rates can approach 800 megabits per second.

The ACTS scheduling system considers a large amount of information from both experimental and operational activities during the scheduling process. This information is classified into four categories: activity, calendar, resource, and state constraints. Activity constraints encompass the requests for duration, terminal usage, bandwidth, rain-fade type, and terminal spot beam location. Calendar constraints include predetermined events such as eclipses of the satellite and planned maintenance. Resources include both the bandwidth constraints for each spot beam and the bandwidth requested by the experimenters. The processors onboard ACTS allow 31 possible configuration "states" connecting uplink beams to the processors then to the downlink beams. Each experimenter requires a subset of these states to successfully complete their experiment.

IMPLEMENTATION STRATEGY

The entire scheduling process begins with a database of user requests. Requests are then individually scheduled by the human

scheduling expert with the aid of the ACTS Scheduler. The generated schedules represent a valid, conflict free set of events that satisfy experimenters' requests. These events are then output in a timeline format that details hour-by-hour events on the satellite. Information is sent through the database which adds domain specific knowledge for configuring the satellite. Configuration orders are then sent to the ACTS Master Control Center to be uplinked to the satellite. This process is shown in Figure 1.

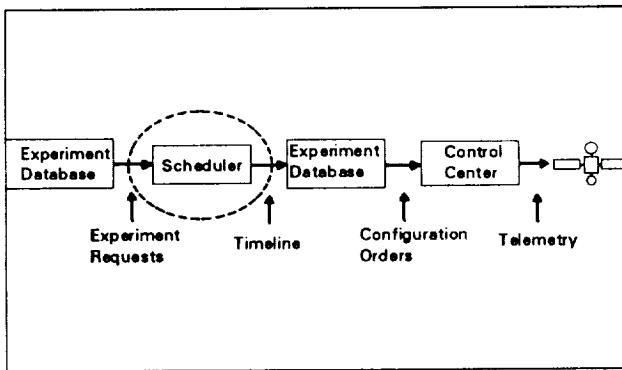


Figure 1 Complete Scheduling Process

SCHEDULING PROCESS

The ACTS Scheduler is a resource-based experiment scheduler [Biefeld 1990, Johnston 1989]. The major resource constraints are classified as capacity (non-depletable) resources which model communication bandwidth. The resource hierarchy must also include parent children relations. A value subscribed to a child resource must also be subscribed to the parent resource, and so on. Because each experiment is usually unrelated to others via temporal relations, temporal precedence constraints are not needed to model the domain of ACTS. Each experiment may request multiple runs, therefore, the ACTS Scheduler must be able to represent multiple instances of an activity. Each of these instances may also be slight variations on the original experiment to meet time and/or

bandwidth constraints during the time frame of the instance.

Schedules are generated in a human-computer interactive paradigm within the confines of a constructive scheduling framework. For reasons that are to detailed to completely justify in this paper, automated scheduling 'rules' are neither necessary nor feasible for inclusion in the ACTS Scheduler. The rules needed for automated scheduling are both difficult to capture and constantly varying. For these reasons, a human-computer interactive paradigm was chosen to generate schedules. In this paradigm, the computer performs all of the computationally intensive valid interval calculations, resource updates, activity instance tracking, while the humans perform the functions that require heuristic knowledge [Fox 1992].

A constructive scheduling framework can be defined in the following manner. The initial schedule is free of constraint violations, being either empty or populated with activities that as a whole violate no constraints. Considering the initial case, the constructive method generates a schedule by 1) choosing an activity to schedule, 2) finding all possible temporal periods that the activity can be placed without violating any constraints, 3) deciding one temporal location to place the activity, and finally, 4) updating all the constraints affected by the activity. This four step process is repeated until either activities can no longer be placed on the schedule (without constraint violations) or no more unscheduled activities exist. In a fully automated scheduling system, items 1 and 3 are the functions that requires heuristic knowledge, while items 2 and 4 require a meticulous and time consuming search and data consistency effort. Items 1 and 3 are often times domain specific, while items 2 and 4 are more generic across multiple scheduling problems. The basis of the joint human computer effort is the split of items 1 and 3 to the responsibility of the human, while items 2 and 4 are the responsibility of the scheduling software.

REPRESENTATION DETAILS

Three extremely difficult representation problems exist within the ACTS scheduling project: unconventional resource hierarchies, multiple admissible state constraints, and context

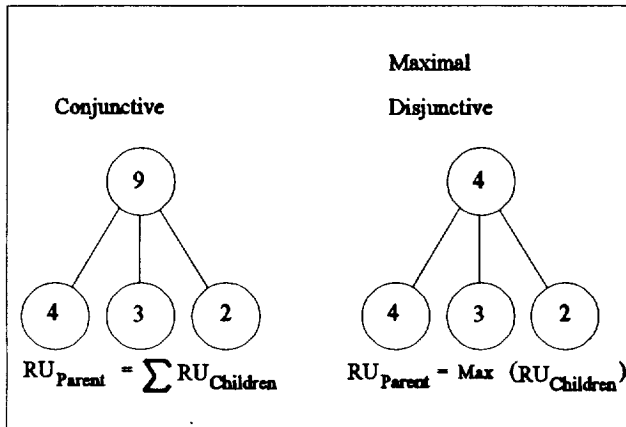


Figure 2 Resource Inheritance Types

dependent overhead. Resource hierarchies are addressed in many commercial scheduling packages, but with a very limited scope. For example, consider a construction scheduling problem where 4 electricians, 3 plumbers, and 2 carpenters are working. In this case, a total of 9 workers are being consumed, the sum of the three specific technical areas. This concept is called conjunctive inheritance. In the ACTS scheduling project, a type of inheritance named maximal disjunctive is defined. The resource usage of the parent is defined as the value of the single largest resource user of its children. For example, if three activities were using 4, 3, and 2 units of a maximally disjunctive resource (which have a common parent), only 4 units would need to be subscribed to the parent resource. These two inheritance types are described in Figure 2. A boolean inheritance is also defined. For each child that consumes a non-zero amount, a value of one (1) is subscribed to the parent. The maximal disjunctive inheritance type is used in the ACTS uplink channels when multiple communication frequencies overlap within the

processing equipment onboard. The boolean inheritance is used to allocate overhead during the sharing of ground terminals.

State constraints are among the most difficult of problems within scheduling. The difficulty stems from the fact that state constrained variables have a temporal cost of transformation from one value to another. In the ACTS scheduling problem, an additional caveat is added, one that I call multiple admissible state constraints. A request for a conventional state constrained variable is in the form *Activity 'a' requests Resource 'r' to be in State 's'*. The multiple admissible state constraints in ACTS can be stated in the form *Activity 'a' requests Resource 'r' to be in one of the States (s_0, s_1, \dots, s_n)*. This adds a host of complications in the representation and reasoning about state resources.

The most unconventional of the constraints in the ACTS scheduler is the context dependent overhead. Since ACTS uses time-division multiplexing, requests for communication bandwidth are actually converted to time slots on the satellite. An activity not only needs multiples of these time-slots, but an overhead amount based upon the number, location, and type of terminals concurrently operating. The rules governing overhead dependency based upon number, location, and type of terminals concurrently operating are not straight forward. Because of the nature of these rules, it is very difficult to incrementally add the correct amount of overhead to the schedule. Therefore, two sets of resource usages are kept, conventional usage and overhead usage. When modifications are made to the schedule, the overhead is recomputed from scratch. If the overall resource usage is needed, these two numbers are simply summed. Another difficulty arises from the fact that the overhead has a temporal extent unrelated to the activity duration. In particular, the overhead allocated to an activity must have a temporal extent that spans the duration between state changeovers.

CURRENT WORK

Operations of the scheduling system started on December 6, 1993. Operations of the satellite have ramped up from checkout phase to an operational phase. During the first few months of operations, a multitude of minor modifications and additions have been completed. All of these additions have been requested by the customer in order to either make the scheduling process run more smoothly or to more correctly model the domain.

Currently, a Graphical User Interface GUI is being developed and tested. Since the ACTS scheduler was developed on such a tight timescale, only a text-based user interface was initially developed. In order to increase the information transfer to the human scheduler, a graphical representation of timelines, resource usages, and Gantt charts is in development. This will allow the human scheduler to more closely and accurately assess the state of the schedule during the scheduling process.

CONCLUSION

The ACTS scheduling project was undertaken with severe time pressures. The software was essentially written in five months with the additional assistance of previous schedulers being written by the author [Ringer 1991, Ringer 1993]. Without the scheduler to generate valid schedules and output them to generate orders for satellite configuration, operations would not have proceeded as smoothly as they have. The scheduler represents a custom designed piece of software that is unavailable in an off the shelf form. Numerous domain specific constraint types have been modeled to accurately solve the scheduling problem. Most importantly, the scheduling system significantly reduced the time necessary to generate and modify valid experiment schedules for ACTS.

REFERENCES

- [Biefeld 1990] Biefeld, E., Cooper, L. *Operations Mission Planner: Final Report.*, JPL Publication 90-16, March, 1990.
- [Fox 1992] Fox, B., *Chronological and Non-Chronological Scheduling*, In Proceedings AAAI-92 Workshop on Knowledge-Based Production Planning, Scheduling, and Control, July, 1992.
- [Johnston 1989] Johnston, M. *Knowledge-Based Telescope Scheduling*. In Knowledge-Based Systems in Astronomy, Springer-Verlag, 1989.
- [Ringer 1991] Ringer, M.J., *Autonomous Power System: Integrated Scheduling*. In *Proceedings NASA Johnson Space Operations, Applications, and Research Symposium*, Houston, Texas, July 1991.
- [Ringer 1993] Ringer, M.J. *Time-Phased Scheduling Abstractions*. Masters Thesis, Case Western Reserve University, May 1993.

Session PS-NT

Planning and Scheduling Workshop: New Techniques

PS-NT.1 Telescope Loading: A Problem Reduction Approach _____	415
J. L. Bresina, Recom Technologies at NASA Ames Research Center, Moffett Field, California, USA	
PS-NT.2 High Performance Techniques for Space Mission Scheduling _____	419
S. F. Smith, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA	
PS-NT.3 Applications of Dynamic Scheduling Technique to Space Related Problems — Some Case Studies _____	423
S. Nakasuka and T. Ninomiya, University of Tokyo, Tokyo, Japan	
PS-NT.4 Solution and Reasoning Reuse in Space Planning and Scheduling Applications _____	427
G. Verfaillie and T. Schiex, ONERA-CERT, Toulouse, France	
PS-NT.5 Empirical Results on Scheduling and Dynamic Backtracking _____	431
M. S. Boddy and R. P. Goldman, Honeywell Technology Center, Minneapolis, Minnesota, USA	
PS-NT.6 Scheduling with Genetic Algorithms _____	435
T. R. Fennel, A. J. Underbrink, Jr., and G. P. W. Williams, Jr., The Boeing Company, Huntsville, Alabama, USA	
PS-NT.7 Integrating Planning and Reactive Control _____	439
D. E. Wilkins and K. L. Myers, SRI International Artificial Intelligence Center, Menlo Park, California, USA	
PS-NT.8 Automatic Generation of Efficient Orderings of Events for Scheduling Applications _____	443
R. A. Morris, Florida Institute of Technology, Melbourne, Florida, USA	

