Astronomical Data Analysis Software and Systems VIII
ASP Conference Series, Vol. 172, 1999
D. M. Mehringer, R. L. Plante, and D. A. Roberts, eds.

TransVERSE: An Architecture for Configuring Astronomical Observations

Gary Curtis, Tom Donaldson, Rob Douglas, Andy Gerb, Mark Giuliano, Natalie Nigro

Space Telescope Science Institute

Abstract. Typically, an observation specified by an observer cannot be directly executed by an observatory. Observation specifications need to be configured so that they can be used by other components of the observatory. This paper motivates the need for configuration tools and describes a software architecture which supports configuring astronomical observations.

1. Introduction

Many software systems have been written for scheduling astronomical observations (Chavan 1998; Johnston & Miller 1994; Kleiner 1999). These systems optimize the scheduling of multiple observations. In contrast, comparatively little effort has been placed in planning and optimizing individual observations. This paper describes a system for preparing and configuring individual astronomical observations. The rest of the paper reads as follows: §2. motivates the need for configuring astronomical observations, §3. outlines a software architecture for configuring astronomical observations, and §4. gives a brief summary of the main results.

2. Motivation

Typically, an observation specified by an observer as a phase 1 or 2 program cannot be directly executed by an observatory. Observation specifications need to be configured so that they can be used by other components of the observatory (e.g., schedulers, commanding, archiving). As described below, the configuration process is more than a simple translation. The process adds information and optimizes the input specification.

Many required activities such as calibrations, data dumps, instrument configurations, and slews are not explicitly specified by the observer. The configuration process expands the user specification, creating an observation plan which contains all the necessary activities to execute the observation. Figure 1 gives an example of adding activities to an observation specification. The example shows a specification with three exposures. Each exposure specifies a filter, a target, and the number of copies of the exposure to be made. The observation plan shows how the configuration process adds activities to the specification. In particular, the process adds a filter move between exposures 1 and 2, creates

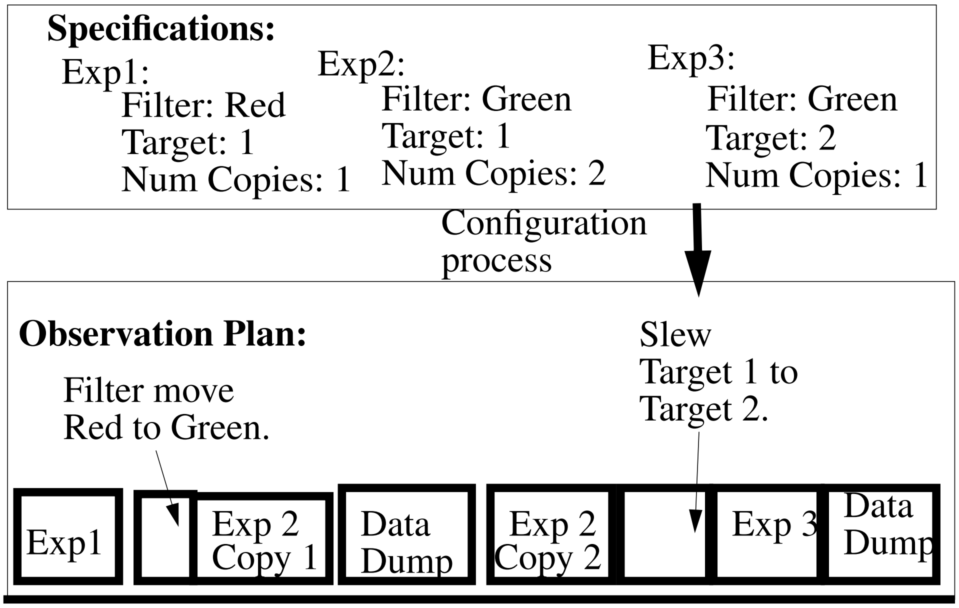

Figure 1. Adding Activities.

two copies of exposure 2, adds a slew between exposures 2 and 3, and adds data dumps.

The configuration system embeds detailed knowledge about how the observatory works so the observer does not need to know these details. As a result, the observer can concentrate on his scientific goals and not the details of how to run an observatory.

In some cases an observation specification only gives ranges of values for input parameters. For example, observations can give ranges of legal times for exposure durations. The choice of actual exposing durations is left up to the observatory and should optimize the use of the telescope. Figure 2 shows a specification with three exposures each with a range of exposure durations. In the example, there are 60 minutes of visibility available for the exposures and the configuration system adjusts the actual exposure durations to fill the orbit.

This approach supports more efficient usage of the observatory. An observation specification defines a set of potential observation plans. The actual observation plan for a specification is selected based on the current conditions available for executing the observation. For low-earth orbiting telescopes, such as the HST, the current conditions include the available orbital visibility. For ground based telescopes, the current conditions include the current weather and seeing conditions.

The configuration process requires the ability to optimize multiple competing objectives. This may require making trade-offs between competing objectives. For example, an observatory might want to:

1. Minimize the duration of observations (i.e., leave more time for other observations).
2. Maximize the time on target (i.e., optimize the current observation).

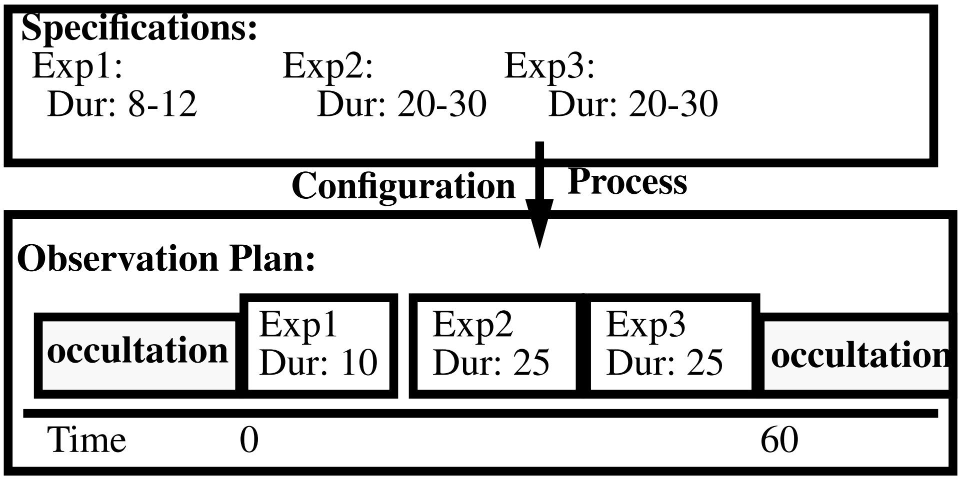

Figure 2. Adjusting Exposure Durations.

3. Minimize overheads (i.e., slews, data dumps, instrument configurations).
4. Place overheads during dead times (i.e., during occultations).

In general, the configuration process requires knowledge of the instrument and ground system in order to optimize multiple competing criteria.

As the use of service mode observing increases, there will be an increased demand for flexible tools for configuring astronomical observations. In classical observing, the investigator actually performs the observations and is free to adjust his plan to meet the current conditions. In service mode observing, the investigator is not available to adjust observations. The investigator must specify any contingencies as part of the specification. A configuration system allows the contingencies to be automatically taken into account.

3. System Architecture

The TRANS system has provided support for configuring HST observations since 1989 (Gerb 1991). The system is being re-engineered under the TransVERSE project. The resulting system will provide a flexible yet powerful tool for configuring astronomical observations.

The TransVERSE architecture supports configuring observations through the following features:

I. The architecture separates input/output functionalities from the process of managing the structure of an observation. Structure management defines the semantics of the system. The input/output modules provide different views of the core semantics. This feature is useful as down-stream systems often require different views of the data.

II. The architecture provides an explicit search model which supports optimizing interacting criteria. Structure management is divided into two components. A search engine generates decisions about the structure (e.g., where to insert calibrations). An object model maintains the current state of an observation, including the implications of search decisions.

III. The object model architecture partitions its knowledge into packages which reason about activities, physical objects, and activity schedules. Activities are entities which are scheduled on the observatory. The physical objects represent components of the observatory and provide the ability to simulate themselves on a set of activities. The system re-uses the temporal constraint propagator, from the SPIKE planning and scheduling system (Johnston & Miller 1994), to reason about activity schedules.

IV. The architecture uses a constraint sequencing infrastructure to automatically track dependencies. On creation of an object, a set of constraints are executed. As the constraints execute, data dependencies are tracked and stored. Constraint enforcement code is automatically executed when a dependency is triggered. This infrastructure frees the developer from having to explicitly sequence the execution of constraints. The constraint sequencer is like a spreadsheet for constraints. A spreadsheet automatically updates the values for cells when dependent cells are modified. Likewise, the constraint sequencing infrastructure automatically executes constraints when dependent values change.

4. Conclusions

The functionality provided by the TRANS system, observation configuration and optimization, has proved invaluable to the efficient and correct use of the Hubble Space Telescope. The job of TRANS is complex as it requires detailed knowledge of the observatory and the ability to optimize competing priorities. The architecture outlined above provides an infrastructure for configuring and optimizing observations for the Hubble Space Telescope and other missions.

References

Chavan, A. M., Giannone, G., Silva, D., Krueger, T., & Miller, G. 1998, in ASP Conf. Ser., Vol. 145, Astronomical Data Analysis Software and Systems VII, ed. R. Albrecht, R. N. Hook, & H. A. Bushouse (San Francisco: ASP), 255

Gerb, A. 1991, Goddard Conference on Space Applications of Artificial Intelligence (NASA Conference Publication 3110), 45, 58

Johnston, M., & Miller, G. 1994, in Intelligent Scheduling, ed. M. Zweben & M. Fox, (San Fransico: Morgan Kaufmann), 391, 422

Kleiner, S. 1999, this volume, 77