Astronomical Data Analysis Software and Systems IX
ASP Conference Series, Vol. 216, 2000
N. Manset, C. Veillet, and D. Crabtree, eds.

The NGST Mission Simulator

S. Speck, R. Samson, S. Stallcup, L. Petro

Space Telescope Science Institute, Baltimore, MD 21218

Abstract. The Next Generation Space Telescope (NGST) is designed to be a successor to The Hubble Space Telescope (HST). The NGST, intended for observations in the near and mid infrared spectrum, consists of an Optical Telescope Assembly (OTA), a sunshield, and an Integrated Science Instrument Module (ISIM).

Reliable estimation of the exposure time and sensitivity of NGST observations is essential to the development of feasible scientific programs. The NGST Mission Simulator (NMS) is a suite of web based tools that is being used for observing program and spacecraft design studies. The NMS calculates exposure times, signal-to-noise ratios, or sensitivities for observations of astronomical targets given a design of the NGST observatory and its potential science instruments. The NMS can also be used to calculate the Mission Elapsed Time (MET) for a suite of observing programs based upon the optimal scheduling of those observations in each program.

This paper describes the current architecture of the NMS and the various scenarios under which the tools can be employed. We conclude with a discussion of future enhancements under consideration. The URL for the main page of the NMS web site is NGST NMS[1].

1. Introduction

The NMS can be used to calculate exposure times, signal-to-noise ratios, or sensitivities for observations of astronomical targets for a specific design of the observatory and science instruments. The NMS can also be used to calculate the Mission Elapsed Time (MET) for a suite of observing programs, based upon the optimal scheduling of those observations in each program. The MET is a primary figure of merit for alternative NGST designs. The NMS employs a model of the signal detection process that accounts for all significant backgrounds (detector, instrumental, observatory, and natural), the throughput of NGST, and the image formation process. The wavelength dependence of all these quantities in the signal detection process is represented in the NMS.

This paper focuses on some of the key architectural components of the NMS. These components were primarily designed with the following objectives: to facilitate rapidly evolving input data and algorithmic changes, to detach the core mathematical functions from the user interface, to make the tools accessible on the world wide web and on individual desktops, and to simplify planned

[1] `http://www.ngst.stsci.edu/nms/main/`

interfaces with external applications. The primary components that make up the NMS architecture are as follows:

1. Plain ASCII text files for defining telescope specifications, instrument characteristics, and observing programs. The files are parameter based files that are subjected to rigorous data validation checks;
2. Command line activated programs for computing exposure times, signal-to-noise ratios, sensitivities, and mission elapsed times. These tools are written in C++ and currently run on the Sun Solaris operating system;
3. A web based client-server interface for accessing the command line tool set. This interface features a client side file input form developed in **HTML** and **JavaScript** and a server side **Java servlet** processor that manages the interactions between the HTTP client requests and the command line programs

2. NMS Inputs

All input data for the NMS reside in format specific flat ASCII files which are accessed locally from disk by the NMS server. These files include:

1. NGST Ephemeris Files - Files describing three dimensional heliocentric NGST position and velocity vectors as a function of time;
2. Telescope/Spacecraft Data - File which contains the specification of the OTA geometric, optical, and thermal characteristics;
3. Integrated Science Instrument Module (ISIM) Data - File containing the specification of proposed science detector and science instrument configurations, including wavelength dependent data;
4. Design Reference Mission (DRM) Data - File which specifies one or more science observation programs, each of which observes a target location using one of the instrument configurations specified in the ISIM File;

3. NMS Command Line Driven Programs

3.1. The MET_CALC Command

The **MET_CALC** command builds a calendar of scheduled visits as specified in a DRM input file. A visit refers to a group of observations which should be scheduled contiguously on the mission timeline. The program uses an exposure time calculator which takes into account the background light from zodiacal dust, the OTA and sunshield self emission, and the instruments. The exposure time calculator also determines various overhead times, including slew times (both major and minor slews), acquisition times (coarse and fine acq), SI setup times, and spectral element setup times. The schedulability of the visits are also restricted by sun and moon avoidance constraints.

The scheduler uses a greedy algorithm, packing the calendar with observations based upon their “scores” at discrete time points during the timeline. The highest score for a particular observation occurs when it schedules most efficiently. Preference is given to observations with high scores. A maximum visit duration can be specified on the command line. If the calculated visit duration exceeds this maximum, single-obset visits can be “split” into sub-visits, each with a correspondingly proportional number of required targets. The final schedule is then displayed to the output report file, along with the list of failed candidates. An instrument level summary is also provided.

3.2. The ESS_CALC Command

The **ESS_CALC** command calculates **E**xposure time, **S**/N, and **S**ensitivity for one or more visits as specified in a Design Reference Mission (DRM) input file. The program uses the same exposure time calculator as the **MET_CALC** command. **ESS_CALC** operates in only one of three modes in a single execution:

1. Exposure Time Mode - exposure times are calculated for each time point in the orbit given a target brightness and S/N for each observation in the DRM;
2. Signal-to-noise Ratio Mode - S/N values are calculated, given a target exposure time, source brightness, OTA and instrument parameters, etc.
3. Sensitivity Mode - Given a target exposure time and S/N value, the minimum observable target flux is calculated, given OTA and instrument parameters.

4. NMS Output Files

The NMS creates flat ASCII output files of differing format and content depending upon both the run mode and the report type specified by the user.

The **ESS_CALC** report file writes exposure time, S/N, or sensitivity data to the output report, depending on the run mode.

MET_CALC writes the schedule timeline as output to the report file, along with the list of candidates which failed to schedule. Report modes can be specified to control the level of detail written to the output file. **MET_CALC** can be run with a user-specifiable flat file called a Parameter Variations File. This causes the program to recalculate exposure times and perform scheduling for each set of input parameters being specified by the user. Varying input parameters enables studies on the effects of changing the spacecraft, OTA, and ISIM characteristics. A similar processing mode uses scale factors, which scale critical input parameters and recalculate the mission timeline with each scaled value. When running with parameter variations or scale factors, the output report contains scheduling results for each set of varied input parameters.

5. Web-based Interface to the NMS

The web-based NMS toolset allows users to run command line driven programs installed on a web server. On the client side, an **HTML/JavaScript** web form viewed from a browser allows the user to select any of the configured input decks, or their own edited plain text file. The user can optionally enter data for a single observation set by filling in the appropriate form fields. The key to this interface is the use of the encoding type **multipart/form-data**, in conjunction with the **method=POST** attribute of the **ACTION** form element. This specification allows user edited files to be sent within the body of the HTTP request files for processing on the server side. Some JavaScript error checks are performed prior to submission of requests.

HTTP Requests are served by a **Java Servlet** application. Servlets are Java programs that extend the functionality of a server, and are run by a Servlet Engine (currently Apache **JServ** in the case of the NMS). Servlets are invoked from a form in a similar manner as CGI scripts. The URL of the Servlet (typically where the Servlet class files are stored) is specified in the **ACTION** attribute of the NMS tool **FORM** element.

A brief note on the NMS Servlet run-time environment. Three working environments are provided: development, test, and operational. The URL value of the **ACTION** attribute is used to specify which set of Servlet classes are to be invoked. In addition, this URL includes encoded text that specifies the location of a control file which defines the location of input, output, and executable files. Furthermore, the URL contains a string that identifies whether the Servlet is to be run in verbose mode, which is useful for debugging. This data, encoded in the URL, is then retrieved by the servlet via the **getParameter()** method of the **HTTPServletRequest** class. As directed, the form fields data are sent via the request input stream using the multipart/form mime type, which are subsequently processed by a Parser class. This mechanism allows data to be sent via the URL and also within the HTTP request file simultaneously.

Once the user data has been extracted from the HTTP request file, temporary NMS input files are stored locally on the server. The Servlet builds a UNIX script which sets up the environment and the command line string based on the user request. Finally, the Servlet executes the script (using the exec() method) and writes the NMS report to the HTTP response file.

The above web interface design clearly breaks down the application into 3 distinct tiers. First, a web form provides the user with a view to the toolset. Secondly, the servlet acts as the intermediary between the GUI and the core NMS programs. The third tier is made up of the NMS programs themselves which encapsulate the NGST models from the remainder of the system. In this configuration, the modifications made to one layer is made as transparent as possible to the other layers.

6. Conclusions and Future Work

The NMS toolset has undergone numerous upgrades since its inception, and has been used extensively in the development of the DRM by members of the NGST community. The current version of this DRM can be accessed from the NMS web site. Several more enhancements are planned, including improvements to the exposure time calculator and the scheduling algorithm, enhanced user interfaces, and better reporting methods. In addition, we are exploring ways to communicate with other on-line applications that would benefit from directly calling the NMS tools. Using the HTTP request and response mechanism, the existing servlet interface could be easily adapted to such needs.

We will also investigate the potential benefits of using XML to standardize the formats of the NMS input files. Finally, using the Java/servlet framework, the use of sessions and JDBC to log user transactions and to maintain state between transactions will also be considered.