Astronomical Data Analysis Software and Systems VI
ASP Conference Series, Vol. 125, 1997
Gareth Hunt and H. E. Payne, eds.

The OPUS Pipeline Applications

James F. Rose[1]

Computer Sciences Corporation, Space Telescope Science Institute, 3700 San Martin Drive, Baltimore, MD 21218

Abstract. OPUS is both a generic event-driven pipeline environment and a set of applications designed to process the spacecraft telemetry at the Space Telescope Science Institute in Baltimore, Maryland. This paper describes those OPUS applications which process the telemetry, validate the integrity of the information, and produce standard FITS (Flexible Image Transport System) data files for further analysis. The applications are to a great extent table-driven in an effort to reduce code changes, improve maintainability, and reduce the difficulty of porting the system. The tables which drive the applications are explained in some detail to illustrate how future missions can take advantage of the OPUS systems.

A science pipeline can be roughly described as a sequence of six basic steps which range from telemetry unpacking to database dredging to data formatting. Under normal conditions turning a raw telemetry stream into a usable science dataset is not even an interesting challenge: you know the input formats, and you know the output formats, and it is often a simple task to understand how to move the bytes around to accomplish that task.

But that's not the whole task. If a pipeline is expected to handle a large amount of data, continuously and robustly, or when that pipeline is designed to run consistently in an unattended environment, it must anticipate problems and take some reasonable action automatically: developers must assume "normal conditions" are the exceptions. Most of the work, and most of the code in such a system is designed to handle the abnormal case.

The OPUS applications which process a telemetry stream into standard FITS files fall into the following six components.

Data Partitioning: This is the front-end workhorse of the telemetry processing. Its function is to scan the incoming telemetry for known patterns and to segment the telemetry stream into its basic constituents.

The OPUS software assumes the telemetry is "packetized"—that is, the stream of data are made up of discrete chunks, and each chunk has a few bytes of information which identifies the beginning of the chunk and some information about the chunk. Chunks can be of different sizes and types. Packets of different types, which can arrive in any order, are segmented and put into their own files.

Here is where most of our "bad data" checks are done. The software asks a number of questions: Can we identify the observation? Can we identify the header? Can we identify the beginning of a packet? Can we tell how long the packet is? Can we tell if the data are in order? Can we determine if any of the

[1]rose@stsci.edu

© Copyright 1997 Astronomical Society of the Pacific. All rights reserved.

data are missing? Do we have enough packets to continue? Do we have a mixed set of packets? etc. In addition to the raw telemetry, two drivers help guide the partitioning process: Telemetry_Patterns, and Telemetry_Locations. Each of these is described briefly in Appendix A.

Data Quality Editing: If telemetry gaps are indicated, this pipeline process constructs a data quality image to ensure the subsequent science processing does not interpret fill data as valid science data.

OPUS must anticipate that telemetry data may sometimes have problems: dropped packets, missing segments, corrupt data. Rather than attempt to "correct" the data, OPUS must simply flag bad or suspect telemetry. This is a much more important issue when dealing with images and spectra than with time-tagged photon events. Data which has an implicit order rather than an explicit time-tag order is susceptible to misinterpretation if whole segments are out of order. In addition to the science packet file and the error indications file, this process also relies on the Telemetry_Patterns table to guide processing.

Support Schedule Keywords: This step produces observation-specific datasets for a single exposure or for any time interval (e.g., an operational shift's worth). Information from proposals/experiments which have been previously logged into a relational database is extracted for the given observations.

By "keywords" we mean the standard FITS[2] (Flexible Image Transport System) definition which allows an eight character keyword name, a value, and a comment. For example:

```
RA_TARG =       215.59415 / right ascension of the target (deg) (J2000)
DEC_TARG=      -12.725682 / declination of the target (deg) (J2000)
```

Keywords and their values, which describe the science, are at the heart of OPUS processing. This information describes the configuration of the instrument, the timing of the exposure, and the existence of associated exposures. It also controls the downstream calibration—that is, all calibration steps beyond the OPUS Generic Conversion determine the parameters of the science exposure from these keywords. Two tables drive this package: Keyword_Names and Keyword_Source.

Data Validation: In addition to the database, another source of keyword information is the telemetry itself. The Data Validation stage actually has two purposes: first to dredge the required words from the telemetry, converting them to meaningful values on the fly, and then to ensure that the actual values and the "planned" values from the database are consistent.

In addition to the telemetry packet files and the support values produced by the Support Schedule Keywords process, four tables drive the data dredging and validation: Keyword_Names, Keyword_Source, Telemetry_Location, and Telemetry_Conversions. See Appendix A below for a brief description.

World Coordinate System: Pointing an orbiting observatory is a well understood problem, but the accurate pointing of a particular aperture for a particular instrument requires some further processing. This task takes information from the database concerning the pointing of the vehicle and, using a table which relates the location of the apertures to the orientation of the observatory, converts the pointing to the standard FITS coordinate system parameters.

---

[2]http://www.gsfc.nasa.gov/astro/fits/fits_home.html

All information required by this process comes from processes upstream: the support values from the Support Schedule Keywords, and the telemetry values from the Data Validation step. In addition the table describing the aperture locations specified by the instrument engineers is accessible in the Aperture_Location table.

Generic Conversion: This process puts it all together, unscrambling the input data, potentially Doppler-shifting the photon events, and constructing FITS files with the appropriate keywords describing the data.

Generic Conversion is the process which converts the unformatted data into FITS files. This involves writing the raw science FITS files as well as the "support" FITS files from the data contained in the dataset: the science packet file, the data quality packet file (if any), the support values, the telemetry values, and the World Coordinate values.

In addition to needing the science packet files and the data quality packet files, this process requires the support values from the Support Schedule Keywords, the telemetry values from the Data Validation step, and the World Coordinate System values. Additionally the following tables drive the system: Keyword_Names, Keyword_Source, Keyword_Order, and Keyword_Rules.

Data Collector: Besides the six basic pipeline steps, a seventh package has been developed to control the flow of exposures through the pipeline. Some calibrations require that a series of exposures be associated with a separate calibration exposure, such as a wavelength calibration. The Data Collector pauses the processing of individual members of an "association" until all members are present in the pipeline.

Appendix A. Drivers

A system which claims to be table-driven obviously requires a number of tables to drive the system. These tables contain the detail which is specific to a particular mission/experiment. Providing the specifications in these tables limits the amount of mission-specific modifications that would otherwise be required.

Telemetry_Patterns: This is a table which describes a hierarchy of telemetry headers, and their relative position and size. Currently the hierarchy is limited to File/Image/Packet/Segment, however extensions are certainly possible where warranted. This table also describes how to distinguish each telemetry stream by looking at the first "few" bytes of the stream.

Telemetry_Location: Within a telemetry stream—especially within the segment, packet, image, and file headers—are words of special interest when interpreting the stream, and interpreting the science. This table describes the location of each mnemonic, or telemetry item, giving its position in terms of byte offsets and bit offsets. A restriction is that each mnemonic must consist of contiguous bits.

Telemetry_Conversions: This table contains parameters for the three types of telemetry conversions that are accommodated within the OPUS system. Discrete conversion assumes the telemetry mnemonic is an index to a table of strings; it is useful for converting a status monitor to "On" or "Off," for example. Piecewise linear conversion is done for simple temperature or voltages. And polynomial conversion is used for more complicated parameters.

Keyword_Names: This table contains the set of keywords for each instrument and for each mode, and information regarding their datatypes, order in the header, default values, and one-line descriptions. This is the essential information required to build the science headers. More complete descriptions of the keywords should reside in an independently maintained keyword database.

Keyword_Source: This table specifies the source of the keyword value. This may be a database relation and field, or a telemetry mnemonic and conversion mnemonic. This table eliminates the need to hard-code database queries, allowing the OPUS system to be more dynamic and more customizable.

Keyword_Order: Keyword Order provides information to enable Generic Conversion to write the header keywords in the correct order in the FITS files. A keyword can only occur once for a particular instrument, but the order of keywords may be different depending on the mode of the exposure. A spectrographic exposure will require a different set of keywords than will a simple image or a more complex time-tagged event product. This table completes the design of the FITS file headers.

Keyword_Rules: Consistent with the OPUS goal of reducing hard-coded algorithms and making the system more table driven, many of the keyword values are derived from others which have already been determined. The keyword rules table employs a simple rule-based parser which allows the value of these keywords to be modified without writing any code. However, complex algorithms to populate a small percentage of keyword values must still be hard-coded.

References

Rose, J., Choo, T. H., & Rose, M. A. 1996, in ASP Conf. Ser., Vol. 101, Astronomical Data Analysis Software and Systems V, ed. G. H. Jacoby & J. Barnes (San Francisco: ASP), 311

Rose, J., et al. 1995, in ASP Conf. Ser., Vol. 77, Astronomical Data Analysis Software and Systems IV, ed. R. A. Shaw, H. E. Payne, & J. J. E. Hayes (San Francisco: ASP), 429

Nii, H. P. 1989, in Blackboard Architectures and Applications, ed. V. Jagannathan, R. Dodhiawala, & L. Baum (San Diego, CA: Academic Press), xix