

OPUS: A Flexible Pipeline Data Processing Environment

Daryl A. Swade
 Computer Sciences Corporation
 Space Telescope Science Institute
 3700 San Martin Drive
 Baltimore, MD 21218
swade@stsci.edu

James F. Rose
 Computer Sciences Corporation
 Space Telescope Science Institute
 3700 San Martin Drive
 Baltimore, MD 21218
rose@stsci.edu

Abstract. OPUS is the automated pipeline processing environment developed at the Space Telescope Science Institute to handle the hundreds of exposures taken by the Hubble Space Telescope each day. This environment, however, is sufficiently generic to be applied to projects as diverse as large X-ray data reduction systems to small low-volume programs with an interest in “lights-out” operations. OPUS is briefly described in this paper which discusses its components, its applications, and its distribution. The OPUS baseline system is now available on CD-ROM for other projects to apply to their own pipeline requirements.

Introduction

The Space Telescope Science Institute (STScI) has developed OPUS as a data processing software system for converting raw telemetry into standard Flexible Image Transport System (FITS) format data files. OPUS is a dynamic, event-driven, distributed processing system that provides an environment designed to handle a large number of observations processed through many steps across a network of computers.¹ OPUS is also an automated system that monitors processing and provides facilities for error identification and repair.

The Hubble Space Telescope (HST) OPUS pipeline has been operational at STScI since December 1995, and the OPUS baseline

system has now been packaged so that other missions can take advantage of this flexible system. OPUS has been recently ported to the UNIX operating system and can currently run on Sparc/Solaris, ALPHA/DIGITAL UNIX, PC/LINUX, VAX/VMS, and ALPHA/VMS platforms. OPUS also supports a pipeline running on any mix of these platforms.

OPUS Components

OPUS can be considered as having three components. First, the baseline OPUS system is a distributed processing platform, independent of the applications that it runs and monitors. Second, the Application Programming Interface (API) is a library of software packages that supports the

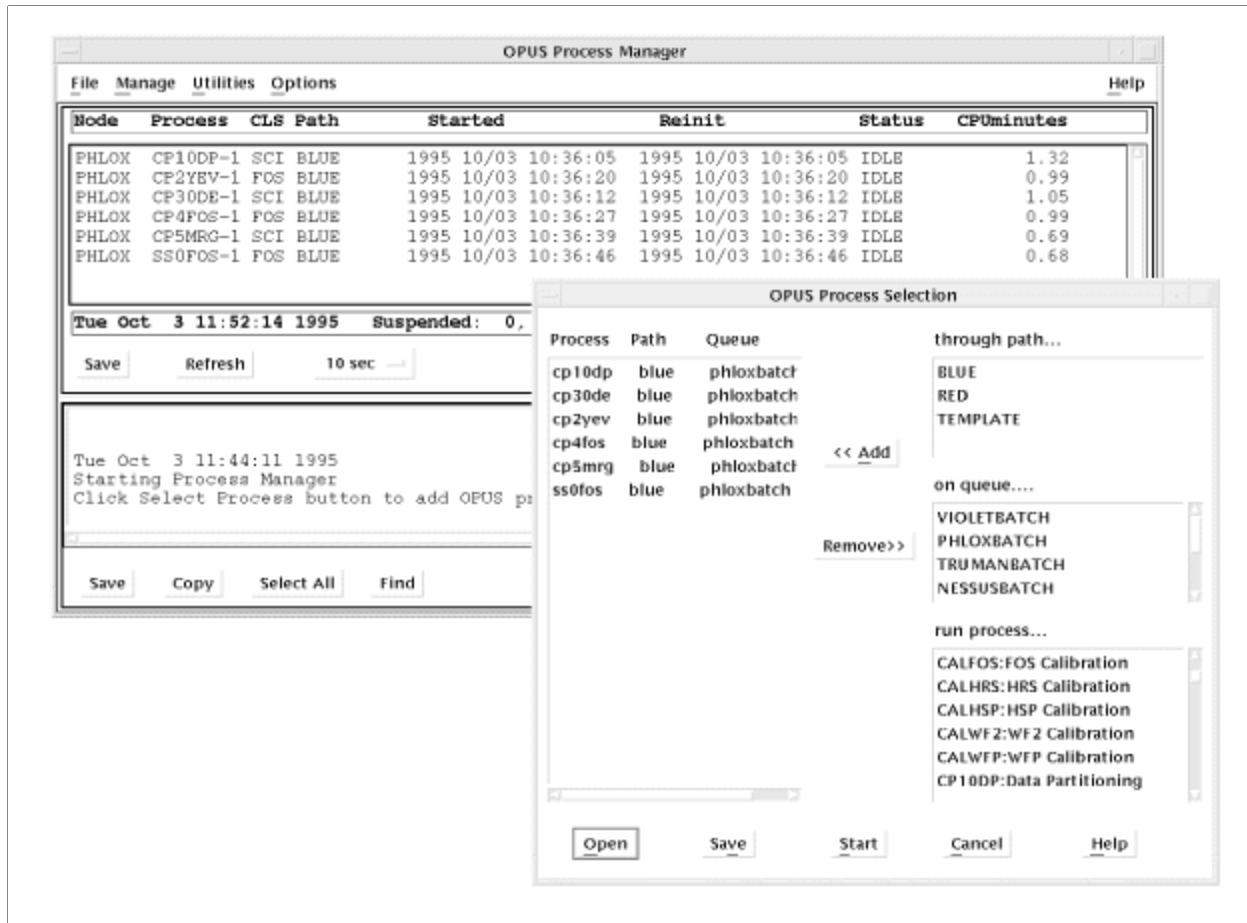


Figure 1: OPUS Process Manager

blackboard architecture for the distributed platform as well as for the standard applications. In addition, OPUS consists of a collection of instrument specific pipeline applications that, until recently, have been developed primarily for the instruments aboard the HST. STScI is currently also developing applications for the Far-Ultraviolet Spectroscopic Explorer (FUSE) mission.²

Baseline System

OPUS has adopted a blackboard architecture using the standard file system of the native operating system for interprocess communication.³ In this model processes do

not communicate with one another, but simply read and write to a common "blackboard" instead of having a single "controller" process which must be continuously aware of the activity of other processes in the system.⁴ This technique effectively decouples the interprocess communication from the individual processes that comprise the data processing system.

Within the OPUS distributed processing system, a variety of independent processes are run sequentially as processing steps in the pipeline. The system also allows multiple instances of any single process to be run simultaneously without interfering

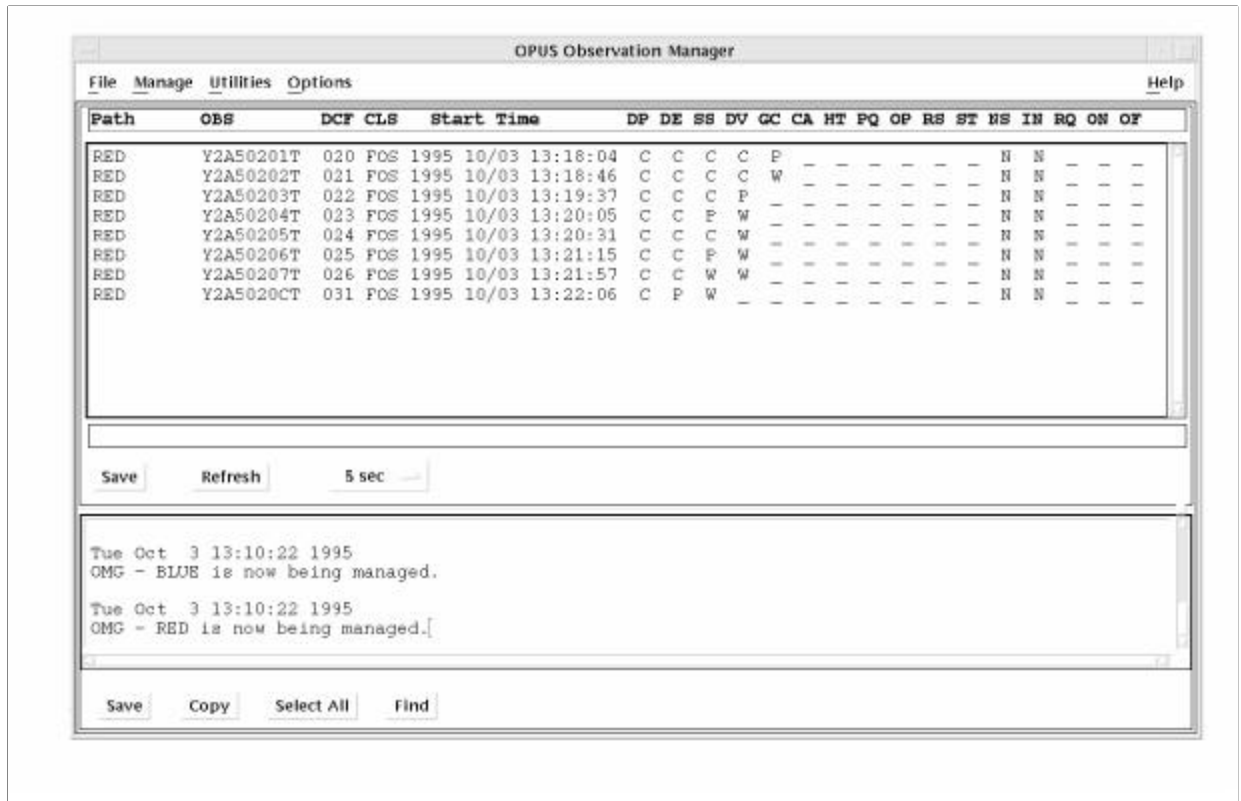


Figure 2: OPUS Observation Manager

with each another. Multiple pipelines, or independent paths, are also supported. Any pipeline can be configured to run its processes across multiple nodes on a network of computers.

In addition to several copies of pipelines with identical processing steps, OPUS supports any number of different pipelines all running on the same network. Thus, in addition to the science pipelines, OPUS can, for example, accommodate an engineering data pipeline and a separate pipeline for other non-science data.

The OPUS environment is configured through a set of simple ASCII text resource files that describe the command line arguments, pipeline triggers, how steps get triggered, and other control information.⁵

The pipeline path file defines a set of cluster-visible directories on the available disks. While one set of disks is being used to process one kind of data, another set can be employed to process a different set of data. Adding an additional machine to the network of nodes is as simple as editing a text file.

In order to monitor the system, OPUS provides two Motif pipeline managers.⁶ The process manager (Figure 1) assists with the task of configuring the system and monitors the status of each process. The observation manager (Figure 2) views the pipeline activities, monitoring the progress of datasets through the pipeline, and flagging observations that are unable to complete pipeline processing. Multiple managers can

each monitor separate pipelines without interference from one another.

OPUS provides facilities for handling data processing problems such as missing data, absent calibration files, or other unexpected situations. The OPUS pipeline provides convenient ways to investigate problems: examine process log files, list data file headers, view observation processing history (trailer) files, and finally restart the troubled exposure at any step in the pipeline.

API

Three categories of support software form the foundation of OPUS and its applications across all operating systems. Utility packages such as string manipulation and time packages are designed to work generically across the various operating systems. Packages that specifically support the blackboard architecture and the operation of the two Motif managers are unique to the OPUS environment. In addition, application packages such as the keyword package, the Open Space Telescope Database (OSTDB), and the FITS++ package⁷ are designed to support pipeline applications directly.

Now that there is a standard C++ language, the utility packages and the basic blackboard system are being upgraded to a true object-oriented environment. This effort will make the entire system more maintainable, more consistent, and more robust. This interface will become the OPUS public Applications Programming Interface (API) and provide software developers direct access to the OPUS blackboards. While that effort is underway, pipeline designers are still able to develop stand-alone applications that know nothing of the blackboard, and use the OPUS environment to tie these applications together into a robust pipeline.

Pipeline Applications

The OPUS applications are programs or scripts that tend to be specific to individual missions or science instruments. OPUS can accommodate an application as any non-interactive shell script. When there is work to be performed by that script, OPUS will pass the name of the dataset to be processed, the location of that dataset and other auxiliary datasets, as well as other parameters required by the script. Similarly the OPUS shell can wrap around any stand-alone, non-interactive executable which takes the name of the input dataset as one argument. All other information for that task is either passed by OPUS in environment variables (symbols) or is obtained from the dataset itself.

Processes (or scripts) can be triggered in three ways: the most common is to allow the completion of one or more previous pipeline steps to act as the process trigger. Another useful technique is to use the existence of a file of a certain class as a trigger. Alternatively, one can use a timing device to trigger an OPUS process (e.g., wake up once an hour).

The OPUS team at STScI has developed a number of relatively generic pipeline applications with an eye to maximize the reuse of OPUS application software for other projects. For example the extraction of engineering telemetry, conversion of that telemetry into engineering units, and packaging this information as keywords or even FITS binary tables is a capability that has been reused for all the HST instruments as well as for the FUSE project.

The generic keyword package is another example of stable and robust software that has been reused to satisfy instrument

specific requirements. FITS format provides a standard so that calibration and analysis algorithms can easily operate on the data. A major component of the output FITS dataset is the keywords defining the data: dataset origin, the configuration of the instrument, the pointing of the telescope, the ambient temperatures of critical components, etc. OPUS allows development of a database of keywords and their definitions. As an example, the HST keyword database was developed at STScI and is publicly accessible through a WWW interface at STScI, www.stsci.edu/archive/keyword.⁸

With OPUS, mission software developers can write specific applications that use the OPUS architecture and API. In that case, experienced STScI OPUS developers are available for consultation and help (opushelp@stsci.edu). Alternately, as with the FUSE mission, OPUS developers can write the application software.

The HST OPUS Pipelines

At STScI separate, but interrelated, OPUS pipelines are used for receiving and processing science telemetry, receiving and processing engineering telemetry, and generating astrometry FITS files. In addition, an OPUS pipeline for processing on-the-fly calibration requests from the Hubble Data Archive is under development.

The OPUS Science Data Processing (SDP) pipeline generates the science data files used in astronomical research, and then queues that data for archiving. For HST, the OPUS science data processing pipeline typically generates over 3 GB of data in about 250 observations per day.

Observatory Monitoring System (OMS) applications use the OPUS system architecture to process HST engineering

data into observation logs for distribution with the science data and files for engineering trending analysis. Observation logs contain jitter and guide star information as well as flags that indicate observational error conditions.⁹ As of March 1998 the OMS system also incorporates processing of astrometry data from the HST fine guidance sensors.

Calibrations for the HST science instruments are continually being improved. For example, observations used to calibrate the HST science observations are typically executed close in time to the science exposures. It may take weeks for the calibration data to be reduced into the calibration reference files used in the pipeline calibration software. Hence, data generated in the OPUS science data processing pipeline a few hours after the observations are executed are not optimally calibrated. Under development at STScI is a system that would re-calibrate data on retrieval from the archive. This on-the-fly calibration (OTFC) system will use an OPUS based pipeline to provide HST archive users with the best possible calibration of the data at the time of retrieval.

HST Science and Engineering Data Flow

The receipt and processing of science and engineering data for the HST utilizes a number of interdependent OPUS pipelines. In addition, these pipelines all rely on a number of common databases. Information from these databases is used to populate header keyword values in the output science and engineering data products, and to monitor and record processing information.

Figure 3 shows a schematic diagram of the HST data processing flow for the OPUS science and engineering data processing

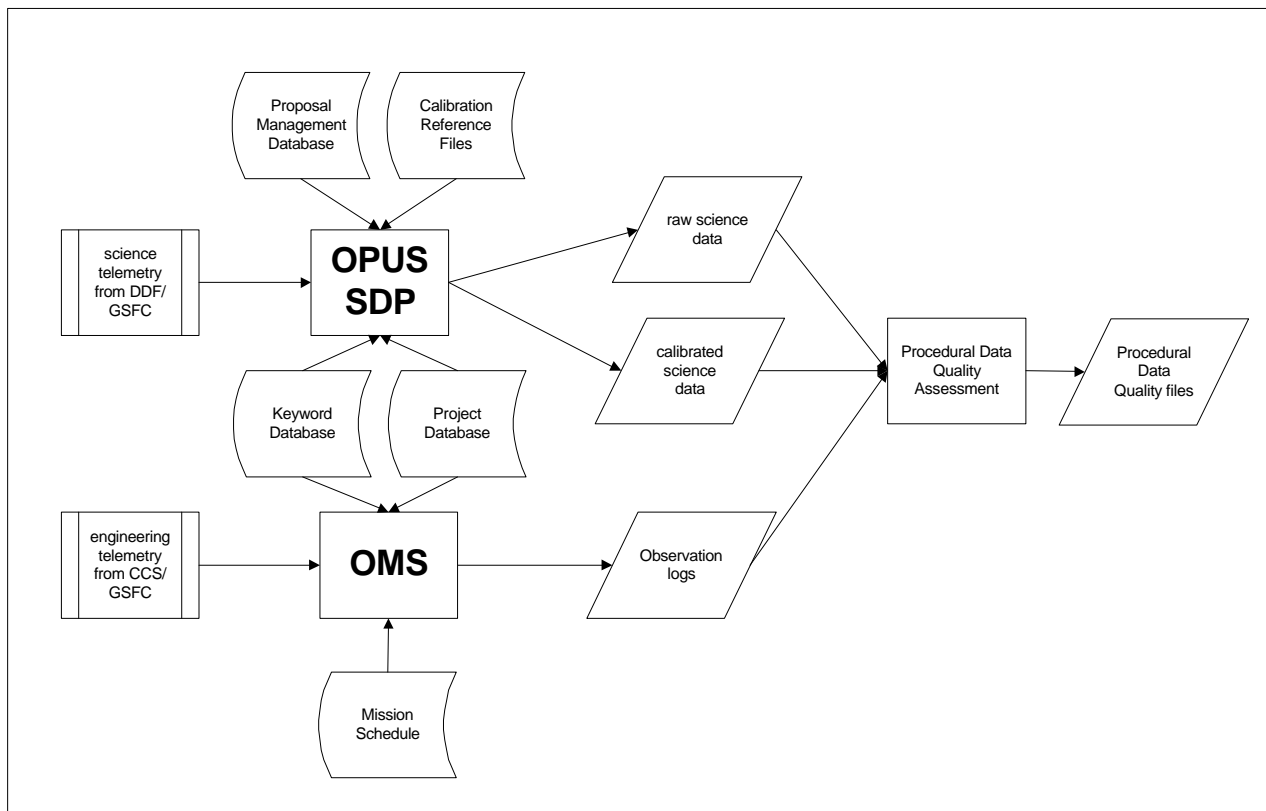


Figure 3: HST Science and Engineering Data Flow

pipelines. Science data are received from the spacecraft by a path that flows through a TDRSS satellite, the TDRSS ground station, a domestic communications satellite, and the Data Distribution Facility (DDF) at Goddard Space Flight Center (GSFC). From DDF, the science telemetry is transferred via FTP to the STScI. At that point, the science data flow through an OPUS data receipt pipeline and are then transferred to the OPUS SDP pipeline.

Engineering data are downlinked from HST along a separate path. In a process similar to science data receipt, the engineering telemetry is transferred from GSFC into the OMS pipeline at STScI. The OMS pipeline uses the OPUS blackboard architecture to sequentially process engineering data. Included in the OMS system are applications

for processing astrometry data from HST's fine guidance sensors.

In addition to the science and engineering telemetry, input for the OPUS SDP and OMS pipelines comes from a number of databases and data files. The Proposal Management Database for HST includes information supplied during the proposal submission process along with information generated from the process of scheduling the proposal to execute on the telescope. A keyword database contains all the information necessary to construct FITS format headers and instructions for the population of the keywords in those headers. The HST Project Database contains mnemonic definitions for downlinked telemetry values and aperture positions. A table in the keyword database links a keyword to the source of its value. In

addition, OMS relies on a mission schedule file that contains a detailed listing of events executed on-board HST and SDP uses reference files to calibrate the individual science instruments.

The output products of the SDP pipeline are instrument specific calibrated and un-calibrated science data files in FITS format. The OMS pipeline generates observation logs that provide information on the behavior of the fine guidance sensors during an observation as well as telescope pointing details. OMS also generates files containing large amounts of engineering data for trending analyses. Additionally, OMS generates astrometry science data files. All data undergo an automated quality assessment process. Data considered suspect by the automated process are flagged for manual inspection.

HST Science Data Processing

Although not included in the distributed version of OPUS, the HST SDP pipeline provides a more detailed example of the specific applications that can be used in an OPUS pipeline.¹⁰ At STScI, an OPUS pipeline is used to receive the data and record accounting information on the telemetry packets. The OPUS science data receipt pipeline passes the packetized telemetry files to the OPUS science data processing pipeline. This pipeline contains a series of processes that convert the science telemetry packets into the astronomical standard FITS output data files.

Although the processes in the SDP pipeline are science instrument dependent, figure 4 illustrates the processes in a typical pipeline. The following processes would execute on each science observation.

Data Partitioning - This process 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.

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.

Support Schedule - This step produces observation-specific information from proposals that have been previously logged into a relational database.

Data Validation - This step decodes the exposure and engineering parameter information in the telemetry and compares these to the planned values.

World Coordinate System - The WCS component implements a translation from telescope coordinates through the instrument light-path to an astronomically valid pointing. The WCS calculations depend on real-time telemetry information.

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.

Data Collector - Some calibrations require that a series of exposures be associated. For example, a separate wavelength calibration exposure may accompany a spectral observation or some number of individually exposed images may be coadded in the pipeline. The Data Collector pauses the processing of individual members of an "association" until all members are present in the pipeline.

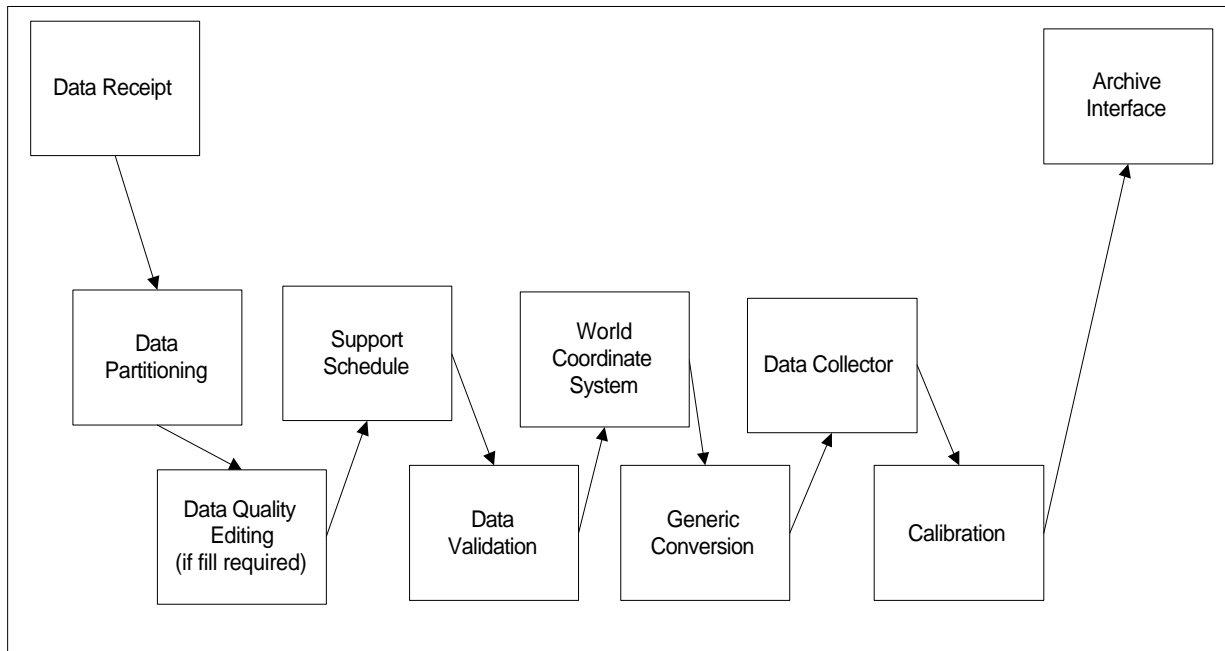


Figure 4: HST OPUS Science Data Processing pipeline

Each observation undergoes an instrument specific calibration. The calibration of HST data is performed by IRAF/STSDAS tasks. These tasks are freely available and can be used by any HST researcher to recalibrate data with updated reference files or calibration algorithms as they become available.

Finally, all data are queued for inclusion in the Hubble Data Archive.

OPUS Distribution

The OPUS baseline system is currently distributed on CD-ROM to help other institutions with their own pipeline management. The OPUS environment was designed to be extensible, and this means that it can be easily modified to work on a variety of projects. The OPUS CD-ROM comes complete with the Process Manager, the Observation Manager, a set of sample

applications, and all the resource files required to get a sample pipeline running.

Sample Pipeline

The Sample Pipeline is distributed with the OPUS environment to demonstrate some of the major capabilities of the OPUS system. This example pipeline simply converts a stack of public GIF images from the Hubble Space Telescope archives into standard FITS files. In the sample pipeline, the GIF images provide the "raw telemetry" files that would normally feed a real data reduction pipeline.

However, more importantly the sample pipeline is a working illustration of how to put together your own production pipeline. Each of the tasks in the pipeline demonstrates a different variety of "trigger" that activates the task. For each task there is a separate text resource file showing the

variety of switches and parameters available to the system. In addition, the pipeline resource files and path files for the sample pipeline are provided in simple text files that you can both examine and modify.

Documentation

The OPUS CD-ROM and the sample pipeline are fully documented in the OPUS Frequently Asked Questions (FAQ). The FAQ is available on the OPUS CD-ROM or at www.stsci.edu/software/OPUS/opusfaq.html. This document explains how to install the system on your computers, how to run the sample pipeline, how each of the Managers works, and what the different resource files are about. As more projects get experience with the OPUS environment, this document expands with further clarifications, and is revised with each new release of the OPUS CD-ROM.

The OPUS CD-ROM does not include the applications that you need to reduce the data for your project. What is included are the pipeline management facilities you need to configure an efficient and robust data reduction pipeline.

Summary

While the OPUS system was developed at the STScI, the blackboard system and the API are independent of the HST mission. HST mission specific applications are not portable, but the experience of the OPUS team in developing complete pipelines for the HST, FUSE, and other potential missions is available. Other projects like INTEGRAL, XMM, MSSSO/Mosaic, WIRE, AXAF and SIRTf are independently investigating and/or tailoring OPUS to their own needs. Packages such as OPUS provide a true resource for NASA and ESA projects in a cost-conscious era where the software

development cycle can and should be better controlled. The OPUS platform and the OPUS software libraries can be reused, forming the basis for the rapid development of robust data processing applications.

References

1. Rose, J., "OPUS-97: A Generalized Operational Pipeline System", in ASP Conf. Ser., Vol.145, 344, Astronomical Data Analysis Software and Systems VII, 1998.
2. Rose, J., et al., "OPUS: The FUSE Science Data Pipeline", in Observatory Operations to Optimize Scientific Return, SPIE, Vol. 3349, 410, 1998.
3. Rose, J. et al., "The OPUS Pipeline: A Partially Object-Oriented Pipeline System", in ASP Conf. Ser., Vol.77, 429, Astronomical Data Analysis Software and Systems IV, 1995.
4. Nii, H.P., "Introduction" in Blackboard Architectures and Applications, eds. V. Jagannathan, R. Dodhiawala, & L. Baum, Academic Press, San Diego, 1989.
5. Boyer, C. and T.H. Choo, "The OPUS Pipeline Toolkits," in ASP Conf. Ser., Vol. 125, 42, Astronomical Data Analysis Software and Systems VI, 1997.
6. Rose, J., T.H. Choo, and M.A. Rose, "The OPUS Pipeline Managers," in ASP Conf. Ser., Vol.101, 311, Astronomical Data Analysis Software and Systems V, 1996.
7. Farris, A. "FITS++: An Object-Oriented Set of C++ Classes to Support FITS," in ASP Conf. Ser., Vol. 125, 262,

Astronomical Data Analysis Software and Systems VI, 1997.

8. Swade, D.A. et al., "HST Keyword Database," in ASP Conf. Ser., Vol. 145, 442, Astronomical Data Analysis Software and Systems VII, 1998.
9. Hyde, P., Perrine, R., and Steuerman, K., "The Observatory Monitoring System: Analysis of Spacecraft Jitter," in ASP Conf. Ser., Vol. 125, 422, Astronomical Data Analysis Software and Systems VI, 1997.
10. Rose, J.F., "The OPUS Pipeline Applications," in ASP Conf. Ser., Vol. 125, 38, Astronomical Data Analysis Software and Systems VI, 1997.

Biographies

Dr. Daryl A. Swade has worked for Computer Sciences Corporation on the HST project for ten years as an Operations Astronomer, Archive Scientist, and Project Engineer. He has been lead of the Data Processing Team since October 1996. He earned a B.S. in Astronomy and Physics from Pennsylvania State University and a M.S. in Physics and PhD in Astrophysics from the University of Massachusetts prior to working at STScI.

James F. Rose, Principal Computer Scientist with Computer Sciences Corporation, Inc., has worked with the Association of Universities for Research in Astronomy (AURA) on the Hubble Space Telescope Science Institute's ground system projects for over ten years. As principal designer of OPUS, Mr. Rose has contributed significantly to the efficient operations of the Institute's production systems.