

The OPUS CORBA Blackboards and the New OPUS Java Managers

Walter Warren Miller III¹, James F. Rose²

Space Telescope Science Institute

Abstract. OPUS is a generic pipeline environment for running multiple instances of multiple processes in multiple paths on multiple nodes. This is the platform which has been used successfully at the Space Telescope Science Institute for over five years to process the HST telemetry. This paper presents the basic concepts of the new OPUS caching blackboards and illustrates these concepts with the new OPUS Java Managers.

1. Background

OPUS³ is a generic pipeline environment that has been used successfully at the Space Telescope Science Institute for over five years to process HST telemetry. It is generic enough to be applicable to other pipeline requirements and has been distributed on CD-ROM to other NASA/ESA observatories free of charge. As a consequence it is now being used by NASA's great observatories (*Chandra*, *Hubble*, and *SIRTF*), and by other observatories throughout the world.

The original OPUS architecture (Rose et al. 1994) was very simple and evolved to meet a variety of needs such as changing requirements and operating systems. A recent review of the original design revealed both the growing complexity of the system and some tremendous opportunities that could enhance the base system to make it more maintainable, extensible, and powerful.

As a result, the OPUS Application Programming Interface OAPI; Miller 1999⁴ was born: an object oriented rework of the original OPUS architecture and a complete and published interface allowing client access to the internal workings of the OPUS blackboards.

2. Caching Blackboards

Riding on the generalization of the OPUS architecture introduced with the OAPI, blackboards can reside anywhere: on the file system, in a database, or in

¹AURA: Association of Universities for Research in Astronomy

²CSC: Computer Sciences Corporation

³<http://www.stsci.edu/software/OPUS>

⁴<http://www.stsci.edu/software/OPUS/OAPI>

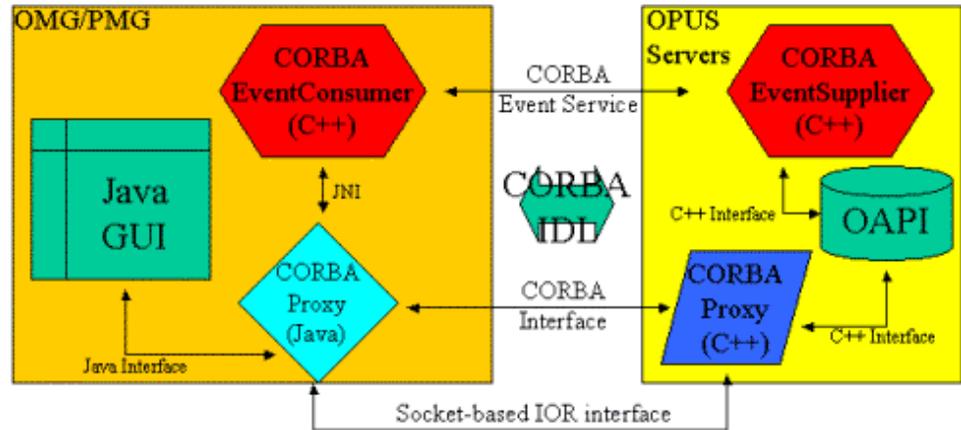


Figure 1. OMG/PMG, CORBA server, and OAPI OPUS interfaces

memory. Instead of using an overworked file server to provide the OPUS cache, a specially designed OPUS server has been developed. The OPUS caching blackboard is an in-memory blackboard (which nevertheless uses the file system as a backing store) where all interprocess communication is provided through a CORBA (Common Object Request Broker Architecture) interface.

Using the new facility of the OPUS caching blackboards to push events, the new OPUS Java Managers (the Observation Manager and the Process Manager) no longer are required to initiate polling. Instead they receive notification from the OPUS Server whenever the blackboard is modified. In consonance with the caching philosophy, a new channel was developed to serve the processes that monitor the OPUS pipelines.

Although a variety of approaches might be taken to incorporate these features in the new managers, CORBA can deliver a homogeneous solution since it promotes both programming language and locational transparency. It offers an object-oriented interface architecturally similar to the OAPI, and includes the desired push event mechanism. CORBA is an industry-standard specification for an object request broker⁵ that acts as a message bus for the transmission of invocation requests and their results to distributed CORBA objects.

The diagram in Figure 1 illustrates relevant interfaces between components in the new managers, between the new managers and the CORBA servers, and between the CORBA servers and the OAPI.

3. Java Managers

Two pipeline managers come with the OPUS system. These are full Java applications that assist the user in monitoring the system. The Process Manager (PMG) not only assists with the task of configuring the system, but monitors what processes are running on which nodes, and what they are currently doing.

⁵<http://www.omg.org>

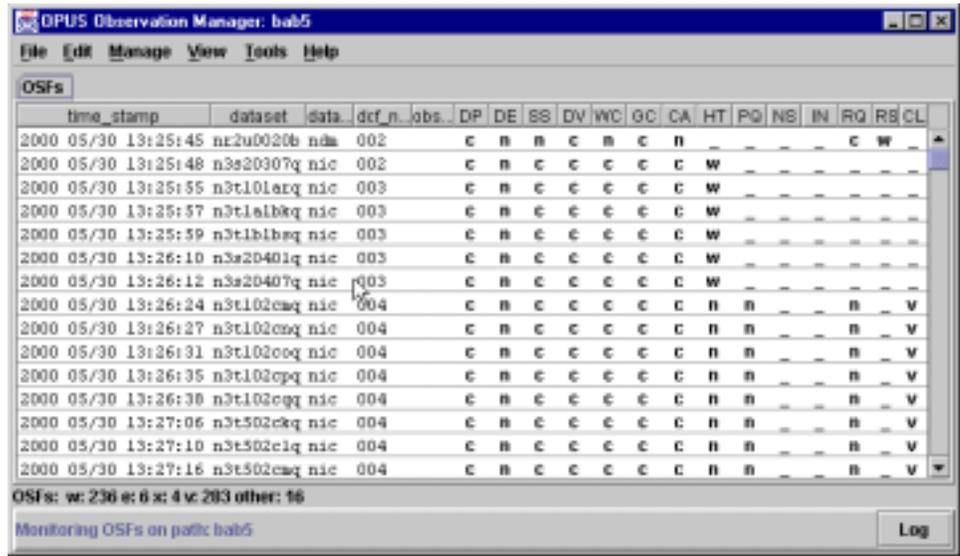


Figure 2. OPUS Observation Manager

The Observation Manager (OMG; see Figure 2) takes a second view of the pipeline activities, monitoring what datasets are where in the pipeline and alerting the operator when observations are unable to complete pipeline processing.

Because the interprocess communication mechanism is CORBA based, the Managers have been written as portable Java applications that communicate with the OPUS blackboards using the standard TCP/IP protocols. This, in turn, frees the Managers from having to operate within the confines of the pipelines they are managing: so long as they have a secure link with the OPUS server, the pipeline monitors can be served anywhere.

Freeing the graphical user interface from the responsibility of tracking events directly resulted in much more responsive tools in the hands of the OPUS user. The OPUS Java managers have taken all the capabilities of the Motif managers they replaced and added important new functionality. The locational transparency afforded by the new OPUS CORBA server implies we can run any number of Managers over the network on our workstations.

The new OPUS Observation Manager continues to support the flexibility built into the original Motif tools, e.g., configurable from external ASCII files, able to handle thousands of entries without loss of responsiveness, and easy modification of the status of any selection of entries. Consistent with the use of the managers as a view into the OPUS blackboards, the GUI provides its own user-defined views into the event lists.

Thus the user can create a new view that displays only the observations associated with a particular instrument, started on a particular day, or having completed a particular step in the pipeline. All views are accessible by clicking on tabs at the top of the display. Full use is made of “tooltips” to expand the use of abbreviations and mnemonics.

The new OPUS Process Manager (see Figure 3) maintains the functionality of the original tools and can monitor many pipelines simultaneously. Like

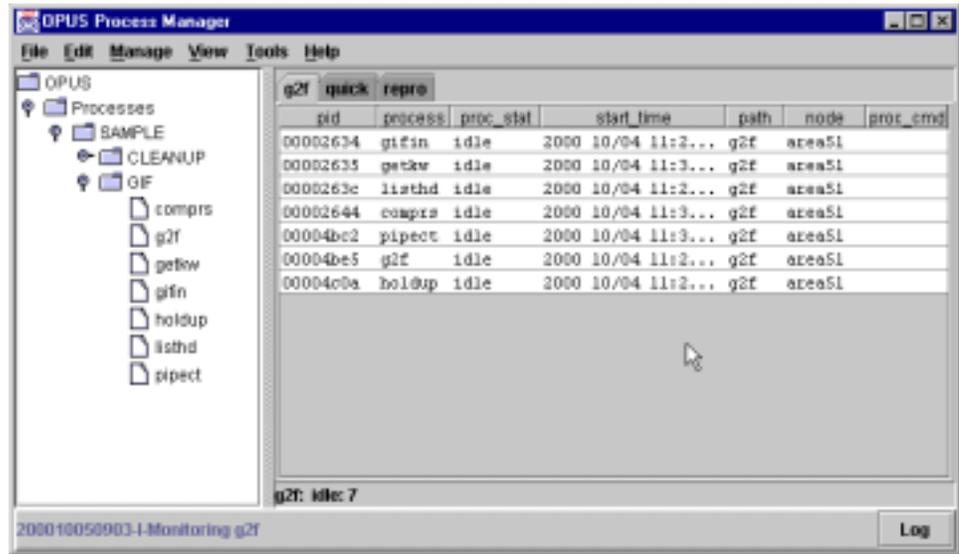


Figure 3. OPUS Process Manager

the new Observation manager that uses the full functionality of the new Java Swingset, the Process manager's display can be easily configured on the fly by each user. Thus all paths defined by the user will be displayed in a separate tab. All OPUS processes are made accessible, and can be dragged to a path to define an operational pipeline.

References

- Rose, J., et al. 1994, 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
- Miller, W. W. III 1999, in ASP Conf. Ser., Vol. 172, *Astronomical Data Analysis Software and Systems VIII*, ed. David M. Mehringer, Raymond L. Plante, & Douglas A. Roberts (San Francisco: ASP), 195