The Chandra Automatic Data Processing Infrastructure

David Plummer and Sreelatha Subramanian

Harvard-Smithsonian Center for Astrophysics, 60 Garden St. MS-81, Cambridge, MA 02138, Email: dplummer@cfa.harvard.edu

Abstract.

The requirements for processing Chandra telemetry are very involved and complex. To maximize efficiency, the infrastructure for processing telemetry has been automated such that all stages of processing will be initiated without operator intervention once a telemetry file is sent to the processing input directory. To maximize flexibility, the processing infrastructure is configured via an ASCII registry. This paper discusses the major components of the Automatic Processing infrastructure including our use of the STScI OPUS system. It describes how the registry is used to control and coordinate the automatic processing.

1. Introduction

Chandra data is processed, archived and distributed by the Chandra X-Ray Center (CXC). Standard Data Processing is accomplished by dozens of "pipelines" designed to process specific instrument data and/or generate a particular data product. Pipelines are organized into levels and generally require as input the output products from earlier levels. Some pipelines process data by observation while others process according to a set time interval or other criteria. Thus, the processing requirements and pipeline data dependencies are very complex. This complexity is captured in an ASCII processing registry which contains information about every data product and pipeline. The Automatic Processing system (AP) polls its input directories for raw telemetry and ephemeris data, pre-processes the telemetry, kicks off the processing pipelines at the appropriate times, provides the required input, and archives the output data products.

2. CXC Pipelines

A CXC pipeline is defined by an ASCII profile template that contains a list of tools to run and the associated run-time parameters (e.g., input/output directory, root-names, etc.). When a pipeline is ready to run, a pipeline run-time profile is generated by the profile builder tool, pbuilder. The run-time profile is executed by the Pipeline Controller, pctr. The pipeline profiles and pctr support conditional execution of tools, branching and converging of threads, and logfile output containing the profile, list of run-time tools, arguments, exit status, parameter files and run-time output. This process is summarized in Figure 1.

Figure 1. The CXC Pipeline Processing Mechanism

3. Pipeline Processing Levels and Products

CXC pipeline processing is organized into different levels according to the extent of the processing. Higher levels take the output of lower levels as input. The first stage of processing is Level 0 which de-commutates telemetry and processes ancillary data. Level 0.5 processing determines the start and stop times of each observation interval and also generates data products needed for Level 1 processing. Level 1 processing includes aspect determination, science observation event processing and calibration. Level 1.5 assigns grating data coordinates to the transmission grating data. Level 2 processing includes standard event filtering, source detection, and grating data spectral extraction. Level 3 processing generates catalogs spanning multiple observations.

4. Standard Pipeline Processing Threads

Figure 2 represents the series of pipelines that are run to process the Chandra data. Each circle represents a different pipeline (or related set of pipelines). Level 0 processing (De-commutation) will produce several data products that correspond to the different spacecraft components. Data from the various components of the spacecraft will follow different threads through the system. The arrows represent the flow of data as the output products of one pipeline are used as inputs to a pipe (or pipes) in the next level. Some pipelines are run on arbitrary time boundaries (as data is available) and others must be run on time boundaries based on observation interval start and stop times (which are determined in the level 0.5 pipe, OBI_DET).

5. Pipeline Processing Registry

The complete pipeline processing requirements for Chandra are very complex with many inter-dependencies (as can be seen in Figure 2). In order to run the pipelines efficiently in a flexible and automated fashion we configure the Automatic Processing system with a pipeline processing registry. We first register all the Chandra input and output data products. We can then capture the processing requirements and inter-dependencies by registering all the pipelines. Data products are registered with a File_ID, file name convention (using regular expressions), method for extracting start/stop times, and archive ingest keywords (detector, level, etc.). Pipelines are registered with a Pipe_ID, pipeline profile name, pbuilder arguments, kickoff criteria (detector in focal plane, gratings in/out, etc.), input and output data products (by File_ID), and method for generating the "root" part of output file names.

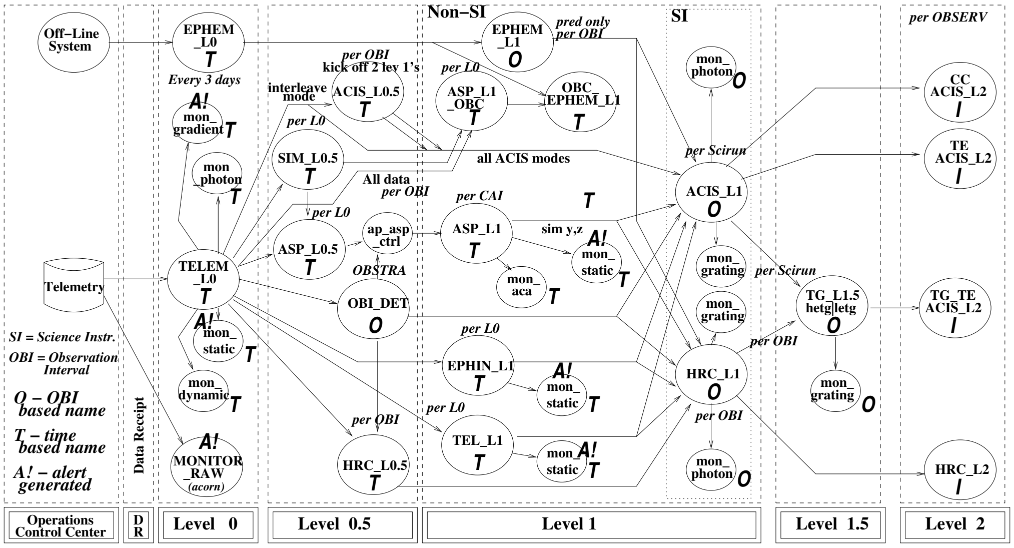

Figure 2. Standard Processing Threads

6. Automatic Processing Components

With a processing registry, the Automatic Processing system is able to recognize data products, extract start and stop times, initiate pipeline processing, and ingest products into the archive. Figure 3 illustrates the flow of data through the AP system. Here is a brief description of each of the AP components in Figure 3:

- The OCC (Operations Control Center) sends scheduled observation and engineering requests, raw telemetry, and ancillary data (ephemeris, clock correlations, etc.) to the CXC.
- Ancillary Data Receipt (implemented via "OPUS") sends ancillary data to the Archive via "darch".
- The Data Archiver/Retriever Server (darch) and the Archive Cache Server (cache) serve as an interface to the Archive. Files sent to darch are first sent to the cache, then the Archive. Darch checks the cache before retrieving from the Archive to save time and reduce the load on the archive. Darch also sends a notification to OST for every data product cached. For more details see the paper in this volume by S. Subramanian.
- Telemetry Data Receipt polls the input directory and picks up new raw telemetry files. It then checks counters and trims off any overlapping data sending the edited raw telemetry file to darch and DR_FlowControl.
- DR_FlowControl sends raw files to the Telemetry Processor one at a time and is used as an entry point for error recovery or reprocessing.
- The Telemetry Processor strips out telemetry into strip files by spacecraft component. It also identifies gaps in the telemetry and the start and stop of observations. The strippers run continuously on a "stream" of raw telemetry. The Extractors can then run on each strip file and decommutate the raw data to create Level 0 FITS files.

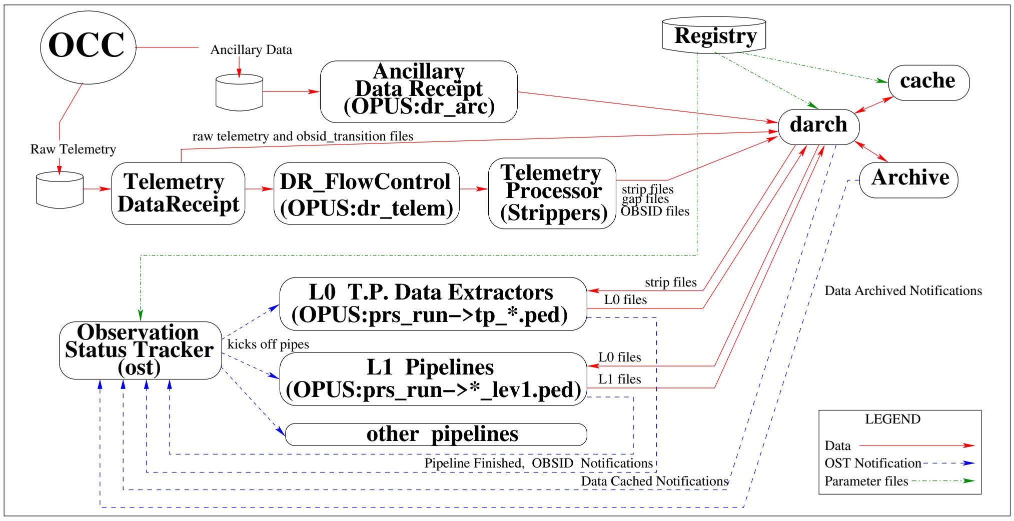

Figure 3. Automatic Processing System

- The OST (Observation Status Tracker) knows about all data products and pipelines via the registry. It sends a message to OPUS to start pipeline processing when all inputs are available.
- The OPUS system is off-the-shelf software from the Space Telescope Science Institute. It provides distributed processing, GUIs for control and directory polling (among other useful utilities). The CXC AP system runs 3 “OPUS Pipes”; Ancillary Data Receipt, DR_FlowControl, and Pipeline Processing. Pipeline Processing consists of 6 OPUS “stages” including data retrieval, running the CXC pipeline, data archiving, notifications, and cleanup. For more OPUS information see the paper by J. Rose (2000).

7. Conclusion

The AP Infrastructure was designed to fulfill a complex set of Chandra processing requirements as efficiently as possible. Instead of hard-coding all the complex requirements and dependencies into software, the AP system relies upon a registry method to configure the processing. The AP infrastructure software can then remain fairly general and maintenance becomes easier as most new processing requirements, enhancements and bug fixes can be accomplished by registry updates. Also, the registry can be easily updated apart from Software Releases for special purposes such as testing, reprocessing and special processing.

Acknowledgments. This project is supported by the Chandra X-ray Center under NASA contract NAS8-39073.

References

Subramanian, S. 2000, this volume, [P2-39]
Rose, J. 2000, this volume, [P2-47]