Astronomical Data Analysis Software and Systems XIII
ASP Conference Series, Vol. 314, 2004
F. Ochsenbein, M. Allen, and D. Egret, eds.

The HERSCHEL/PACS Common Software System as Data Reduction System

Ekkehard Wieprecht

Max Planck Institut fuer extraterrestrische Physik, Garching/Germany

J. Brumfit, J. Bakker, N. de Candussio, J.J. Matthiew, S. Ott, H. Siddiqui

ESA - Astrophysics Division, Noordwijk/Netherlands

R.Huygen, B. Vandenbussche, W. de Meester

Institute of Astronomy K.U. Leuven, Leuven/Belgium

S. Guest

Rutherford Appleton Laboratory,Chilton/United Kingdom

A. de Jonge, P.Zaal

Space Research Organization Netherlands, Groningen/Netherlands

S.Osterhage, M.Wetzstein, E.Wiezorrek

Max Planck Institut fuer extraterrestrische Physik, Garching/Germany

Abstract. ESA's Herschel Space Observatory to be launched in 2007 with a planned lifetime of three years, is the first space observatory covering the full far-infrared and submillimeter waveband (60 -670 microns). By probing so much further into the infrared than any other space telescope, it will have the potential to discover the earliest proto-galaxies and to clarify how they evolved. The Photodetector Array Camera and Spectrometer (PACS) is one of the three science instruments. It employs two Ge:Ga photoconductor arrays and two bolometer arrays to perform imaging line spectroscopy and imaging photometry in the 60 - 210 micron wavelength band.

The HERSCHEL Ground Segment is based on a common, object oriented system - the Herschel Common Science System (HCSS), implemented using JAVA technology and an object oriented Database (Versant). We present the PACS Common Software System Architecture as part of the common Ground Segment. The PACS Common Software System (PCSS) base on the HERSCHEL Common Software System (HCSS) written in a common effort by the HERSCHEL Science Center and the three instrument teams.

1. Introduction

The HERSCHEL Ground Segment Data Analysis Software System base on the programming language Java. This allows a close connection to the object oriented Database System (Versant[1]). For HERSCHEL data analysis an interactive environment is required, where user can work on the command line, write their own scripts and use GUI interfaces. On the other hand highly automatic systems like Quick Look Analysis (QLA) and Standard Product Generation (SPG) shall use plug in components, as they are developed and used within the interactive environment.

In order to fulfill these requirements, the data analysis software system is based on four layers :

- As first layer Jython was selected to support interactivity with Java.
- The second layer is the Data and Operation layer which defines simple Data (e.g. arrays) and basic Operations, which do not require units or error handling, in a user friendly manner.
- The third layer is the Dataset and Algorithm layer. Data are organized in Datasets like Tables. Algorithms can be applied to Datasets considering error handling and units.
- The last layer is the Product and Process layer, wrapping Datasets and Algorithms and responsible for history handling. Products and Processes are managed by DataFlows.

The layers concept is fundamental for all three instrument data analysis systems. On top of it the instrument specific data definitions and applications are built. The goal is to keep the data analysis systems of the three instruments as long as possible in this environment. This has advantages, like a common look and feel for users or maintenance aspects; but also disadvantages like overheads in the common development, increased complexity, and introduced dependencies. To what extent the common approach is applicable and efficient is under discussion.

2. The Java/Jython Layer

To support interactivity and script-ability within Java the programming language Jython[2] is used as the base for the HERSCHEL data analysis system. Jython is an implementation of the high-level, dynamic, object-oriented language Python seamlessly integrated with the Java platform. Jython is freely available for both commercial and non-commercial use and is distributed with source code.

Jython is especially suited for the following tasks:

- Embedded scripting - Java programmers can add the Jython libraries to their system to allow end users to write simple or complicated scripts.

---

[1] `http://www.versant.com`

[2] `http://www.jython.org`

- Interactive experimentation - Jython provides an interactive interpreter that can be used to interact with Java packages or with running Java applications.
- Application development - Python programs are typically 2-10 times shorter than the equivalent Java program. It is also possible to inherit directly from Jython code to benefit your Java applications.

2.1. The Data and Operation Layer

The purpose of the Data and Operation layer is to provide an easy-to-use set of numerical array classes and common numerical operations, for use in Interactive Analysis of Herschel data.

Numerical Operations are provided as function objects that can be applied to array objects. This separation of concerns allows the user to define new functions without having to modify the array classes to add new methods.

The layer provides a Jython-friendly interface, so that interactive users can apply function objects in the form: y = f(x). Functions inherit the necessary Jython wrapping from a base class, so that it is not necessary to write individual wrappers for each function.

It supports array expressions, in which functions and arithmetic operations are implicitly mapped over the elements of an array without using explicit loops.

3. The Dataset and Algorithm Layer

The Layer of Datasets provides a meaning to the Data in terms of annotation, Meta Data and Quantity. The generic base are : Array Datasets, Table Datasets and Composite Datsets.

The Algorithms are higher order operations that may take care of error calculation, unit conversion and attributing the Data. Algorithms are applied on Datasets. Datasets can be saved to files (FITS, XML) or to the Database.

3.1. PACS specific Instrument Level Test Datasets

PACS uses the TableDataset as base for science data, as they are produced during instrument tests or as they come from the satellite, without any processing (except standard unpacking telemetry packets and decompression).

Pacs Spectrometer Raw Data are the raw spectrometer ramps. A time field indicates the start of the ramps, the second column holds the detector identification and the third the actual ramp. This Dataset is used for data originated from various systems : Instrument level test data, Sub-Unit test data of detector arrays and scientific instrument simulator data.

The Pacs Spectrometer Reduced Data are the actual result of on board data reduction (or ground simulations and tests). Again the time serves as index key to get the association with the raw ramps.

The third Dataset holds additional information about the status of the instrument.

Of course these are very basic Datasets, used for first instrument tests. More advanced DataSets are on the way. For example the ImageDataset or SpectrumDataset.

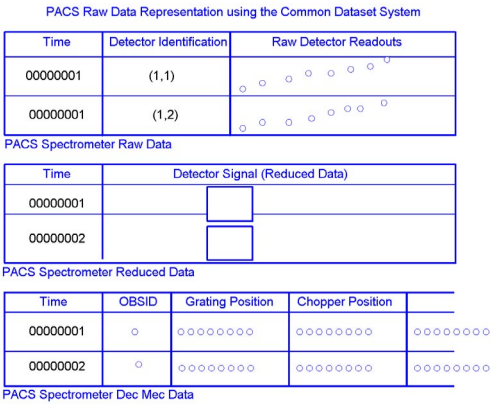

Figure 1. PACS Instrument Test Datasets

4. The Product and Process Layer

A Product is a clone-able data-object, always linked to one unique history. The Product is used as an input/output of a Process as well as a Process control (for example: calibration data). Products can be saved to files (FITS,XML) or to the Database.

Three different types of Products can be defined:

- Official Products (HCSS) : Capable-persistent and fulfills all History requirements Using its History it is possible to reconstruct the DataFlow from which the Product arises from, including the persistent Products (inputs and/or controls) which where used as input for this DataFlow
- Tainted Products : These Products have a History, but automatic DataFlow reconstruction is not guaranteed
- Non-trace-able : Products without History

4.1. The Data Flow Manager and PACS QLA

The DataFlow Manager is a graphical Environment to control Processes. Processes and related input, output and control Products can be controlled purely via the GUI of the DataFlow Manager. It is used e.g. for controlling the DataFlow of the PACS Quick Look Analysis System (QLA). Also the QLA systems of the other instruments are using this application..

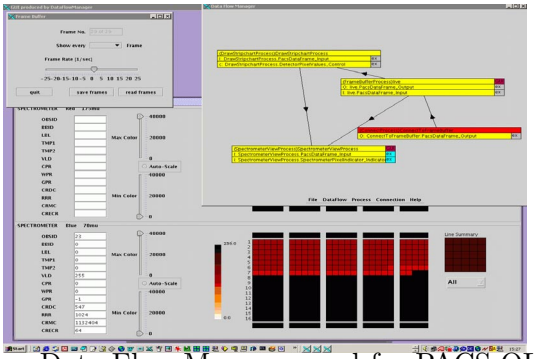

Figure 2. Data Flow Manager used for PACS QLA