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

Implementation Design of the ASC Data Model

J. Herrero, O. Oberdorf, M. Conroy, J. McDowell

AXAF Science Center, Smithsonian Astrophysical Observatory, 60 Garden Street, MS 81, Cambridge, MA 02138

Abstract. The ASC data model provides an abstract description of *AXAF* datasets. Through the data model API, tools and applications will have efficient and transparent access to heterogeneous disk formats. To accomplish this, the data model will use a layered design.

1. Requirements

Previous requirements for the ASC data model stressed flexibility, extensibility, and economy (Conroy et al. 1995). These requirements can be further quantified so that an implementation design can be easily judged. What follows is a list of the quantified requirements that the implementation design had to meet:

Open Architecture: The system shall be implemented so that support for new data formats or new functionality can easily be added, without disrupting previous code or already developed tools and applications.

Standard API Independent of Data Format: The system shall be implemented so that different disk formats can be accessed through the same API.

Code Reuse: The system shall be implemented atop existing libraries that will do the actual low level file I/O.

Support ASC Needs: The system shall support the need of the ASC (*AXAF* Science Center) data analysis environment, so that tools and applications can easily be developed.

2. Implementation Approach

In order to to meet the requirement of support for multiple file formats through the same API, the library uses a layered approach. The bottom layer provides a standard API to multiple file formats. This API provides the other levels with a set of generic objects, such as tables and images, and will be implemented by adopting the ETOOLS package currently under development by a collaboration of CEA and SAO (Abbott et al. 1995; Abbott et al. 1997). The middle layer will add “meaning” to the ETOOLS data objects so that ASC applications can be designed easily and implemented in an consistent way. It will also provide scientific functions to manipulate the data. The top layer will consist of the

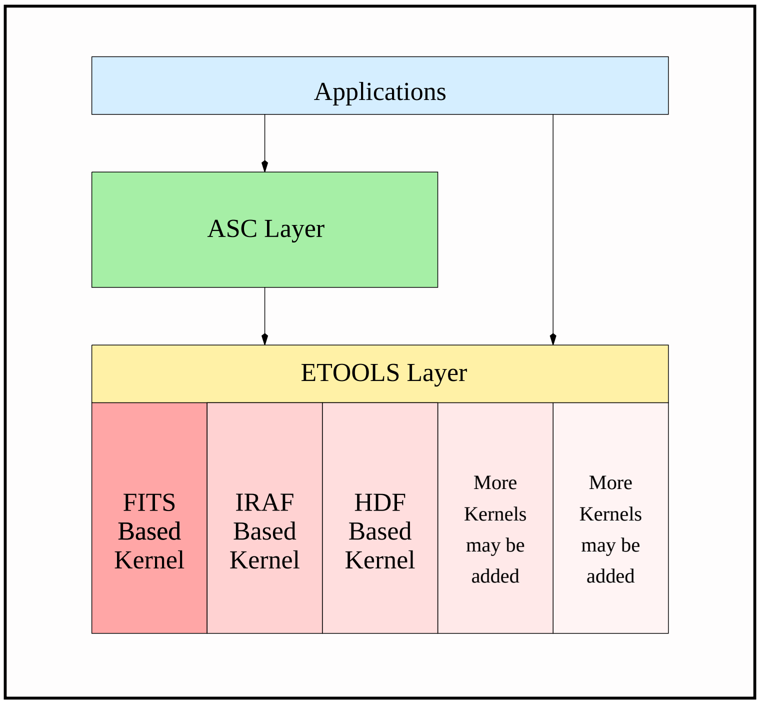

Figure 1. Layered Design of the ASC Data Model.

actual ASC tools and applications. Please refer to Figure 1 to see a diagram of the different layers in our implementation design.

3. ASC Layer

The ASC Data Library layer will provide the required *AXAF* science functionality to the generic data. It will be responsible for providing scientifically meaningful data objects, constructed from ETOOLS primary constructs, and data manipulation functions.

3.1. Dataset Layer

The Dataset Layer gives scientific meaning to the underlying generic objects. This layer will enhance or group the generic objects to produce specialized objects. For example, tools that need to operate on a "Light Curve" object would go through this interface to access their source or target objects. Here are two examples of new objects that the Dataset Layer would provide:

ASC Light Curve Object consists of a table with a binned time column and other columns containing either count rate or flux information and other auxiliary information. It is compatible with HEASARC rate files, and is interpreted as a light curve of count rate or flux versus time.

ASC Source List Object consists of a table with a column including positional information, a column containing source name identifiers (optional), and other optional columns. It is interpreted as a list of sources, and is used as input to programs which calculate derived source properties.

3.2. Science Layer

The Science Layer lies just above the Generic Layer. It contains routines to add additional meaning and functionality to the generic objects. For example, this layer will allow a column to have associated error values. Some examples of Science Layer functionality:

Error Columns: The library will keep track of columns which are error values for other columns in a table object. Science Layer function calls will automatically detect and handle cases where error values are provided.

Coordinate Transforms: The library will keep track of values which are actually coordinates. Science layer function calls will automatically perform any required coordinate transformations.

4. Generic Layer

The decision to adopt the ETOOLS package as the generic layer of the implementation of the ASC data model was made because it meets the implementation requirements. The most important way in which ETOOLS meets these requirements is by providing a uniform interface through which different data file formats can be accessed. This is done by a kernel mechanism, in which there exists one kernel for each data file format desired (e.g., one kernel for FITS files, one kernel for IRAF files, etc.). The following list outlines all the ways in which the ETOOLS package meets the ASC data model implementation requirements:

Open Architecture: The ETOOLS kernel mechanism allows more file formats to be supported by adding more kernels.

Standard API Independent of Data Format: Because of the ETOOLS kernel approach, tools and applications designed and implemented on top of the ETOOLS package can access different disk file formats through the same API.

Code Reuse: The ETOOLS project is not inventing any new file formats. Instead, it provides a standard way to use existing file formats and libraries.

Support ASC Needs: The ETOOLS project provides enough primary constructs to support the ASC tools and application needs, including support for tables, images, filters, and WCS transforms.

4.1. Supported Kernels

The ASC will write at least two ETOOLS kernels to implement its Data Model, the first will be an IRAF based kernel and the second kernel will be a FITS based kernel. Other kernels could be implemented, like an HDF based kernel.

IRAF Based Kernel would use IRAF formats to store the disk data. Event tables will be stored using QPOE files, images will be stored using IRAF image files, and filtering would be done by the QPOE/QPEX filter library.

FITS Based Kernel would use FITS to store the disk data. Event tables will be stored as FITS table extensions, images will be stored as FITS images, and filtering will be done by an ETOOLS provided filter engine.

HDF Based Kernel would use HDF file formats to store the disk data. At the moment this (possible) kernel has not yet been designed.

4.2. Kernel Example, an IRAF QPOE Kernel

The ASC IRAF kernel will use IRAF files to store data on disk. One requirement for this kernel is that the object files (and especially the event tables) can be copied out of the ETOOLS environment, and still be valid IRAF files that can be analyzed in the IRAF environment.

Tables are one of the more important objects in the analysis of event data. Event tables will be stored using QPOE files.

Images will be implemented using IRAF image files.

Filtering will reuse the power and flexibility of the QPOE/QPEX filtering mechanism.

Object Properties will be implemented using either QPOE headers or image headers, depending on the situation.

Acknowledgments. This project is supported by NASA contract NAS8-39073 (ASC). We also gratefully acknowledge the many fruitful conversations with the other members of the ASC Data System and Science Data System.

References

Abbott, M., Kilsdonk, T., Olson, E., Christian, C., Conroy, M., Brissenden, R., & Herrero, J. 1997, this volume, 96

Abbott, M., Kilsdonk, T., Olson, E., Christian, C., Conroy, M., Brissenden, R., Van Stone, D., & Herrero, J. 1996, in ASP Conf. Ser., Vol. 101, Astronomical Data Analysis Software and Systems V, ed. G. H. Jacoby & J. Barnes (San Francisco: ASP), 57

Conroy, M., Doe, S., & Herrero, J. 1996, in ASP Conf. Ser., Vol. 101, Astronomical Data Analysis Software and Systems V, ed. G. H. Jacoby & J. Barnes (San Francisco: ASP), 285

Van Stone, D., Conroy, M., & McDowell, J. 1996, in ASP Conf. Ser., Vol. 101, Astronomical Data Analysis Software and Systems V, ed. G. H. Jacoby & J. Barnes (San Francisco: ASP), 199