Astronomical Data Analysis Software and Systems V
ASP Conference Series, Vol. 101, 1996
George H. Jacoby and Jeannette Barnes, eds.

The ASC Scientific Data Model and its DDF Implementation

D. Van Stone, M. Conroy, J. McDowell

Smithsonian Astrophysical Observatory, Cambridge, MA 02138

Abstract. The AXAF Science Center (ASC) is defining a standard scientific Data Model for use in its data analysis system. This Data Model is an abstract description of underlying data structures, with components such as lists, arrays, header values, and filters. The design of the implementation includes the use of the Dynamic Data Format (DDF) which provides a uniform API (i.e., the Data Model) to the developer, yet allows the user the flexibility of multiple disk formats. Corresponding to each disk format is an API and its own model of the data. The ASC Data Model requires additional functionality and more structure than any individual Off-The-Shelf (OTS) model provides. This paper describes both the components of the Data Model and the design of a flexible layering scheme for implementing the DDF.

1. ASC Data Model

The purpose of a data model is to provide a scientific interpretation of datasets by defining abstract data structures and access functions independent of the physical storage formats (Farris & Allen 1992). This distinguishes between information that is truly part of the scientific data and information that is bookkeeping or specific to the file format, and hence, it is easier to support multiple file formats with the same data model. The data model has a corresponding Application Programming Interface (API) which can access and manipulate the data using the concepts of the data model.

The data model which will be central to ASC data processing will consist of two types of datasets: arrays and tabular lists. An array will be referred to as an ASC image, and a tabular list will be referred to as an ASC table. An ASC image consists of three components: header information, data volume, and image data. The ASC table consists of a header, a data volume, and its table data.

1.1. Header Information

The header is an ordered collection of header data. Header data consist of a name, a data type and a data value. These data can optionally have comments, units, legal range (lower and upper limit values), display format, and coordinate transforms. The data value can also have an associated uncertainty. Some of the header data are marked as fixed header data. In an ASC Table, these data are considered to be additional columns of the table which have a fixed value for the entire dataset. For an ASC Image, these data are considered as separate fixed axes of the image data. Header data will also be grouped for easier manipulation.

1.2. Data Volume

The Data Volume is a mathematical description of the volume which has been applied on the dataset's attributes. Here, the attribute may be either a table column name, an image axis name, a fixed data name, or a fossil name (i.e., one which is no longer in the dataset, but is present to reflect the selection from which the data was made). The scientific importance of the Data Volume is to keep along with the data a description of the range of data values to which the dataset applies. The purpose of the Data Volume is to unify the treatment of good time intervals, spatial regions, and filter ranges, making these concepts independent of the actual attribute.

1.3. Image Data

The image data is an n-dimensional array of a fixed type. The array is considered to have n axes and one pixel type. Each pixel value can have an associated uncertainty. These axes and the pixel values may also have a name, a legal range, comments, units, and display formats, just like header data. Each axis may have a coordinate mapping from integral bin numbers to some scientifically meaningful value.

1.4. Table Data

A table consists of a matrix of rows and columns, with each column containing the same type of data. If a column consists of an array of data, the array size must be the same between arrays in the same column. Each column must have an associated name. It may also have a legal range, comments, units, and a display format.

2. Data Model and DDF Layers

The goal of the DDF design is to provide a uniform API to the user, yet allow the flexibility of multiple disk formats (Conroy et al. 1995). Since application software uses the abstract interface via the API only, the same software may be used to access any of the supported physical formats. Each disk format is associated with a set of library routines. In this paper, these libraries will be referred to as Untouched Off-The-Shelf (UOTS) software.

The general DDF design is shown in Figure 1. In addition to the UOTS wrappers for existing physical formats, there will be an internal representation of the Data Model, with the same interface as the UOTS wrappers. This internal storage allows the library to encapsulate the tradeoff of loading the data fully into memory before manipulation, versus relying on the UOTS software to perform the action.

There may still be requirements on the Data Model functionality which can not be met by the UOTS wrappers. Any differences will be designed into the top Data Model layer. Duplicated functionality between the top layer and an

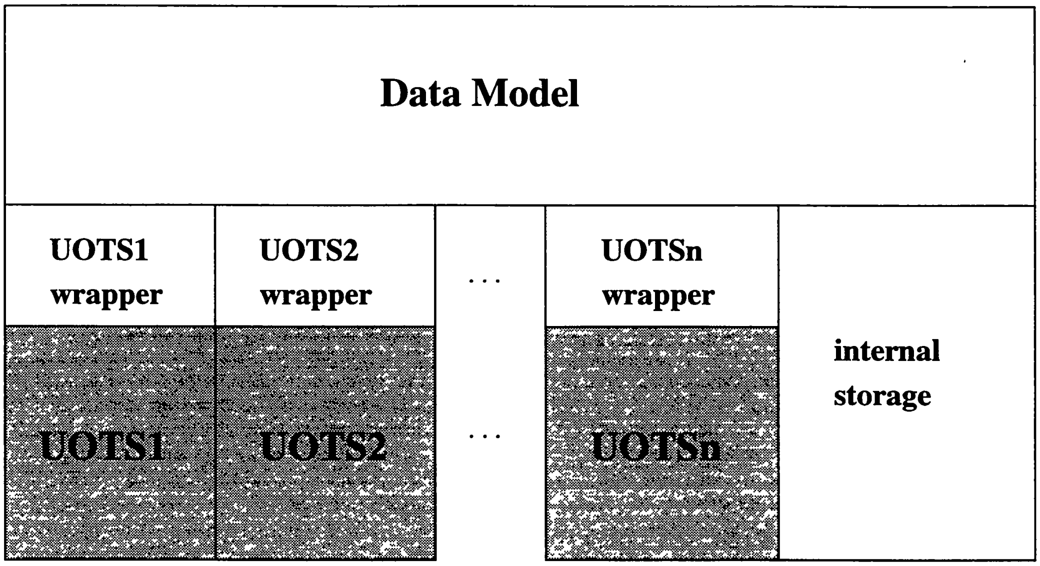

Figure 1. Data Model and DDF Layers.

individual UOTS layer can be encapsulated as a tradeoff at the Data Model layer.

The separation of software into component layers achieves the goal of system flexibility as well as analysis environment independence. Changes within a supported physical format or extensions to future formats can be accommodated by adding or modifying the specific UOTS wrapper within the library, transparent to the rest of the analysis applications that use the Data Model API.

Figure 2 shows the specific UOTS software upon which the Data Model will be layered. One set of wrappers will be written to support the PROS-style QPOE files written during ASC calibration testing. The FITSIO software will provide support for FITS images and binary tables. Another UOTS wrapper for the DDF will be the Event Data Format (EDF), currently being developed by the ETOOLS project, which will allow basic QPOE files to be read and written (Abbott et al. 1996). Single ST Tables will be supported with a wrapper on top of the TBTABLES libraries. This provides the capability to read and write text tables, since TBTABLES supports both binary and text tables. IRAF images will be supported with a layer on the IRAF IMIO library routines. This design allows for additional UOTS software support, if needed. Furthermore, if there are specific efficiency needs not met by other formats, the design allows for additional formats to be added to the design.

Acknowledgments. This project was partially supported by contract SA305-26304PG (ETOOLS) and the NASA contract NAS8-39073 (ASC).

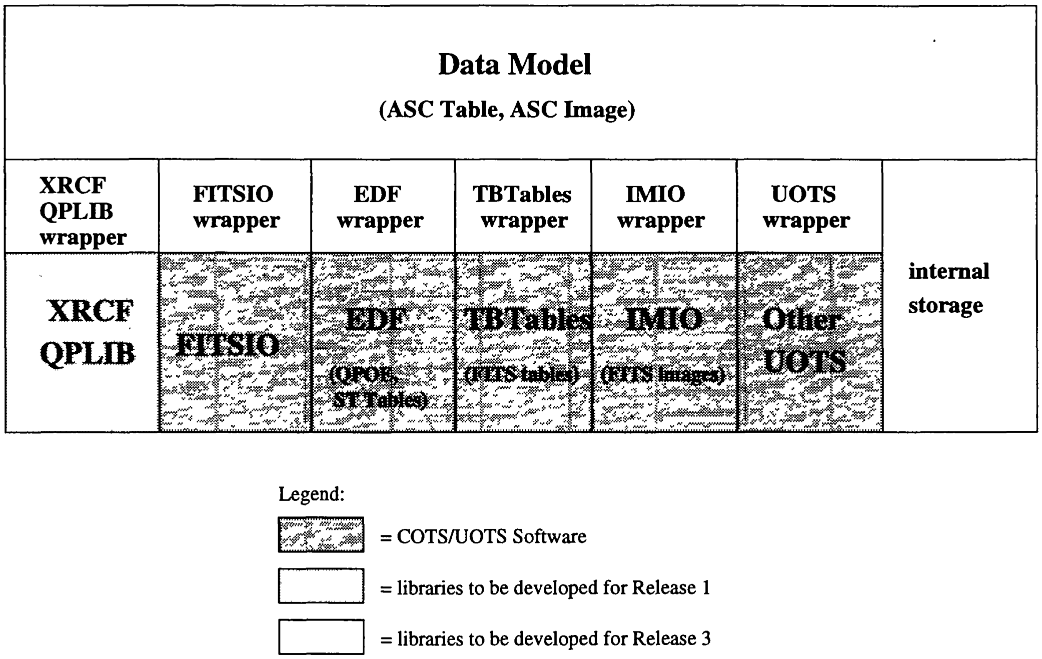

Figure 2. Data Model and DDF Layers.

References

Abbott, M., Kilsdonk, T., Christian, C., Olson, E., Conroy, M., Brissenden, B., Van Stone, D., & Herrero, J. 1996, this volume

Conroy, M., Simon, R., McDowell, J., & Barry, K. 1995, in Astronomical Data Analysis Software and Systems IV, A.S.P. Conf. Ser., Vol. 77, eds. R. A. Shaw, H. E. Payne & J. J. E. Hayes (San Francisco: ASP), 207

Farris, A., & Allen, R. J. 1992, in Astronomical Data Analysis Software and Systems I, A.S.P. Conf. Ser., Vol. 25, eds. Diana M. Worrall, Chris Biemesderfer & Jeannette Barnes (San Francisco: ASP), 157