Astronomical Data Analysis Software and Systems VII
ASP Conference Series, Vol. 145, 1998
R. Albrecht, R. N. Hook and H. A. Bushouse, eds.

The ASC Data Archive for the AXAF Ground Calibration

P. Zografou, S. Chary, K. DuPrie, A. Estes, P. Harbo and K. Pak

Smithsonian Astrophysical Observatory, Cambridge, MA 02138

Abstract. A data archive is near completion at the ASC to store and provide access to AXAF data. The archive is a distributed Client/Server system. It consists of a number of different servers which handle flat data files, relational data, replication across multiple sites and the interface to the WWW. There is a 4GL client interface for each type of data server, C++ and Java API and a number of standard clients to archive and retrieve data. The architecture is scalable and configurable in order to accommodate future data types and increasing data volumes. The first release of the system became available in August 1996 and has been successfully operated since then in support of the AXAF calibration at MSFC. This paper presents the overall archive architecture and the design of client and server components as it was used during ground calibration.

1. Introduction

The ASC archive is projected to contain terabytes of data including ground and on orbit raw data and data products. The archive stores the data following requirements for data distribution across sites, secure access, flexible searches, performance, easy administration, recovery from failures, interface to other components of the ASC Data System and a user interface through the WWW. The architecture is extensible in order to accommodate new data products, new functions and a growing number of users.

2. Data Design

Data such as event lists and images need to be kept in files as they are received. They also need to be correlated with engineering and other ancillary data which arrive as a continuous time stream and to be associated with a calibration test or an observation ID. A level of isolation between the data and users is desirable for security, performance and ease of administration. The following design was chosen. Files are kept in simple directory structures. Metadata about the files, extracted from file headers or supplied by the archiving process, is stored in databases. This allows file searches on the basis of their contents. Continuous time data extracted from engineering files is also stored in databases so the correct values can be easily associated with an image or an event list with defined time boundaries. In addition to partial or entire file contents, file

© Copyright 1998 Astronomical Society of the Pacific. All rights reserved.

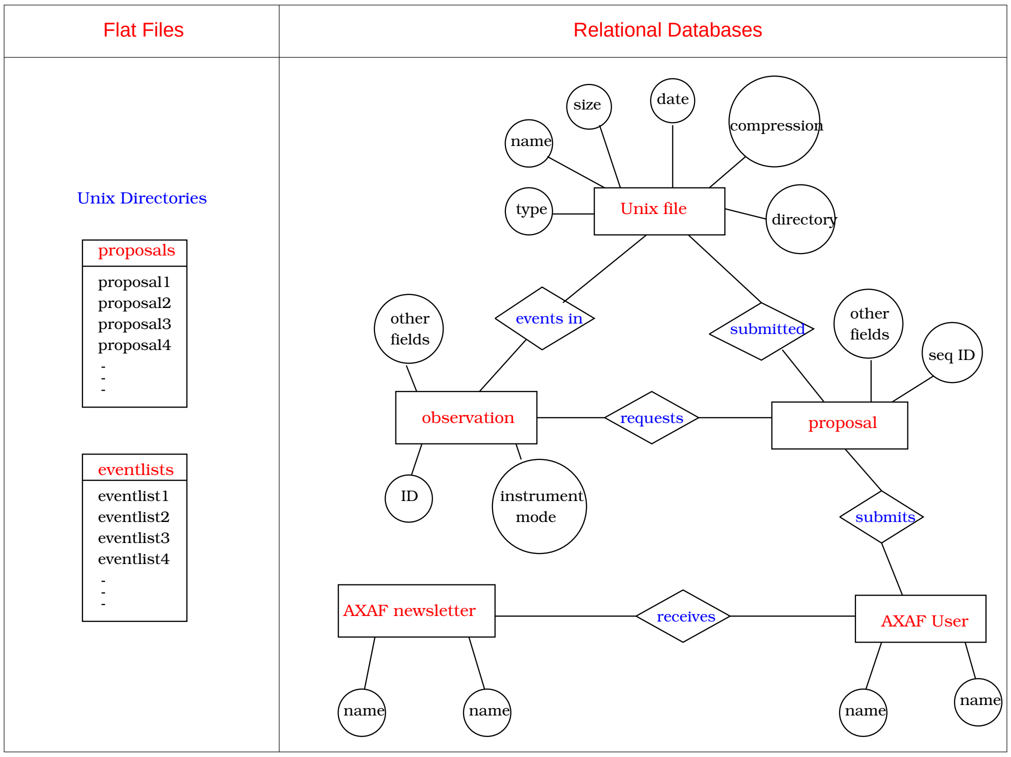

Figure 1. Data Design.

external characteristics such as its location in the archive, size, compression, creation time are also stored in databases for archive internal use. In addition to databases with contents originating in files, there are also databases which are updated independently, such as the AXAF observing catalog and the AXAF users database.

A simplistic example of the data design is shown on Figure 1. The archive contains a number of proposal files submitted by users. It also contains a number of event files, products of observed data processing. A table in a database contains the characteristics of each file. The proposal table contains a record for each proposal which points to the associated file in the file table. The observation table contains a record for each observed target and has a pointer to the associated proposal. An observation is also associated with files in the file table which contain observed events. Related to the proposal is the AXAF user who submitted it and for whom there is an entry in the AXAF user table. An AXAF user may have a subscription to the AXAF newsletter.

3. Software Design

The data is managed by a number of different servers. A Relational Database server stores and manages the databases. It is implemented using the Sybase SQL Server. An archive server was developed to manage the data files. The

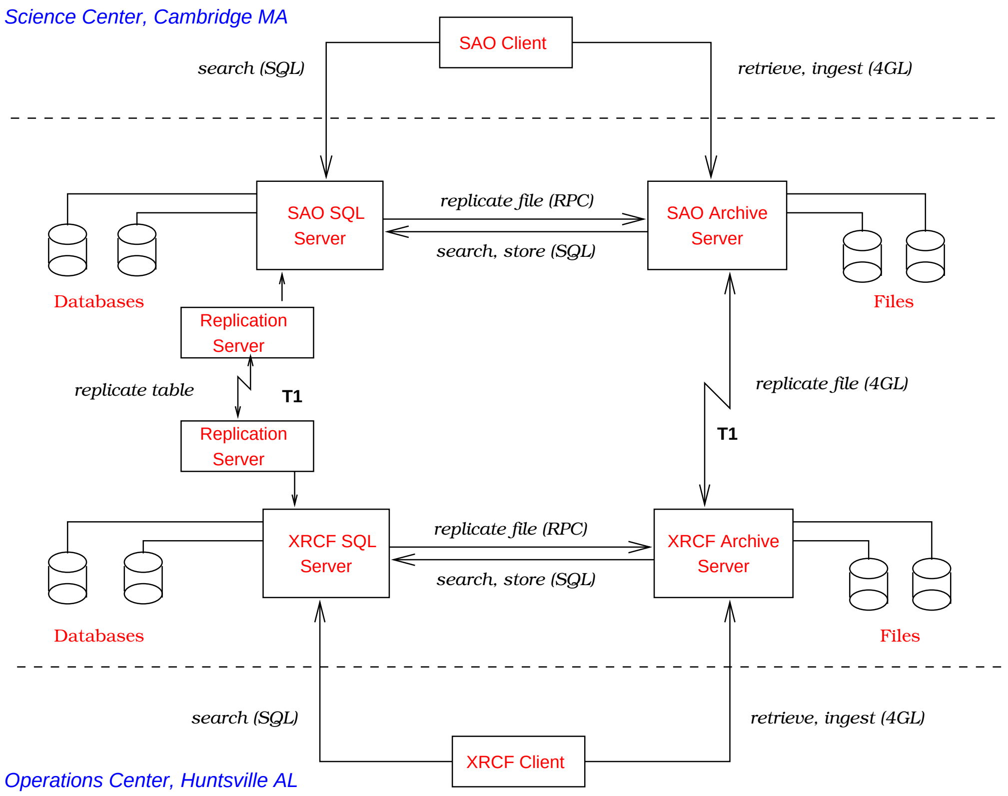

Figure 2. Server Configuration for XRCF Calibration.

archive server organizes files on devices and directories. It keeps track of their location, size, compression and other external characteristics by inserting information in a table in the SQL Server when the file is ingested. It also has data specific modules which parse incoming files and store in databases their contents or information about their contents. The server supports file browse and retrieve operations. A browse or retrieve request may specify a filename or enter values for a number of supported keywords such as observation or test ID, instrument, level of processing, start and stop time of contained data. Browse searches the database and returns a list of files, their size and date. Retrieve uses the same method to locate the files in the server's storage area and return a copy to the client. The archive server responds to language commands and remote procedure calls. Language commands are used by interactive users or processes in order to archive or retrieve data. A custom 4GL was developed in the form of a "keyword = value" template which is sent by clients and is interpreted at the server. The remote procedure call capability is used for automated file transfer between two remote servers.

The server infrastructure uses the Sybase Open Server libraries which support communications, multi-threading, different types of events and call-backs and communications with the SQL server. A C++ class layer was developed to interface the libraries with the rest of the system (Zografou 1997). File transfer uses the same communications protocol as the SQL server which is optimized for data transfer and integrates with other server features such as security.

A third type of server was needed in order to automatically maintain more than one copy of the data at two different locations. The Sybase Replication Server is used to replicate designated databases. Via triggers in the database at the target site the local archive server is notified to connect to its mirror archive server at the source site and transfer files. Queuing and recovery from system or network down-time is handled entirely by the Replication Server.

Client applications use the Sybase Open Client libraries with a custom C++ interface (Zografou 1997). The same client libraries are used for client applications to either the SQL or the archive server.

4. Configuration for Calibration at XRCF

During ground Calibration at the X-Ray Calibration Facility at MSFC two archive installations were operating, one at the operations site at XRCF and a second at the ASC. Communications across sites were via a T1 line. Each installation consisted of a SQL Server and an archive server. A set of replication Servers were setup to replicate all databases which triggered replication of all files. The system layout is shown on Figure 2. Data the in form of files entered the system at XRCF, which was the primary site, and was replicated at SAO. With some tuning to adjust to unexpectedly high data rates the system kept up with ingestion, replication and retrievals by processing pipelines at XRCF and users at the ASC. There were no incidents of data corruption or loss and the overall system was very successful.

5. Conclusion

At the end of the XRCF calibration the system was adapted to support ASC operations at the SAO and AXAF OCC sites connected with a T3 line. In the new configuration only critical data is being replicated. All other data is distributed according to user access. A new server component, the Sybase JConnect Server, and a new Java/JDBC client interface have been added to support WWW access (Chary 1997). The second release of the system, including the WWW interface, is currently operational in support of proposal submission.

References

Zografou, P. 1997, Sybase Server Journal, 1st Quarter 1997, 9

Chary, S., Zografou, P. 1997, this volume