

Eugene F. Burger^{1*}, Jason F. Fabritz¹, and Nancy N. Soreide²

¹JISAO, University of Washington, Seattle, WA 98195

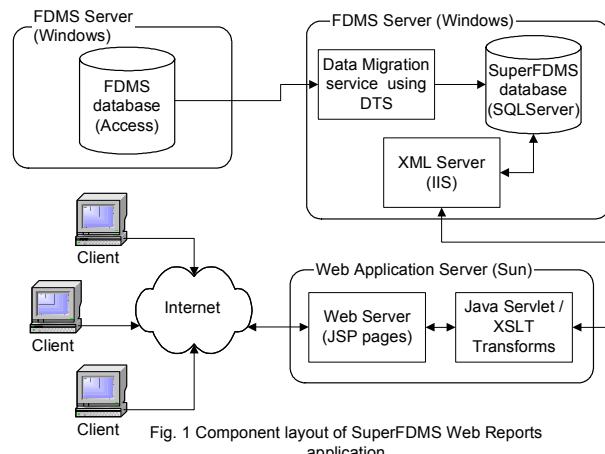
²NOAA/OAR/PME, 7600 Sand Point Way NE, Seattle, WA 98115

1. INTRODUCTION

The OAR Financial Data Management System (OAR FDMS) has been developed to meet the needs of the 18 laboratories and program offices in NOAA Research. Implemented in Microsoft Access, this internal system is hosted on a server located at PMEL in Seattle, and used by the administrative staff in the NOAA Research laboratories and program offices throughout the country with CITRIX Metaframe. Administrative staff input budget and administrative information using Microsoft (MS) Access forms and queries and web forms, and can view status of all current data while on-line. NOAA Research offices each manage their own FDMS data, which are housed in an MS Access database.

The SuperFDMS project was initiated in order to provide NOAA Research headquarters management with consolidated financial and project information across all the operating units in the organization, and is shown schematically in Fig. 1. This paper describes the technologies being utilized in development of the SuperFDMS.

To enable the centralized reporting required for the SuperFDMS, all data are migrated daily from the 18 separate Access databases to a single MS SQLServer database.



* Corresponding author address: Eugene F. Burger, NOAA/PME/OCRD 7600 Sandpoint Way, Seattle, WA 98115; e-mail: eugene.burger@noaa.gov

The SQLServer database is then used by the reporting utilities, including the SuperFDMS and local PI Reports (Fabritz, 2002). The project has four distinct components:

- Data migration from the distributed Access databases to the new consolidated database.
- The new centralized database (Microsoft SQLServer 2000).
- A HTTP/XML server to serve as middleware between the Web server and the Database server.
- Web backend software with web server

2. TECHNOLOGY CHOICES

2.1 Platform

Early in this project we decided not to implement all of the components of this implementation on the same platform. The two platforms used are Microsoft Windows 2000 Server edition, and Sun Solaris 8.

2.2 Database

The database considerations were Oracle 8i and MS SQLServer 2000. The considerations here were the application features, cost, scalability and ease of data migration from legacy databases. Since the next version of the FDMS application will run from a single database server (as opposed to 18 Access databases), scalability was a very important consideration.

We decided to use the MS SQLServer 2000 standard edition database server. This database has all the fundamental features we require (transaction based queries). The Data Translation Service (DTS) feature included with this database is a big advantage of this database as it allows rapid development of data migration applications from Access to SQLServer. SQLServer also proved to be considerably less expensive when compared to other Database servers using similar pricing models. SQLServer allows us to use some of the legacy client software when the FDMS application is converted from Access to SQLServer.

2.3 Web Application Server

We decided to use Apache running on a Sun server since there is a substantial knowledgebase on this

configuration at PMEL. The server software is written in Java and uses the Tomcat servlet engine and JSP parser.

3. IMPLEMENTATION

3.1 *Data Migration*

Data migration was implemented using the Data Transformation Package of MS SQLServer 2000. The scripts are written in JavaScript and extract data from the separate Access databases into the SQLServer database. The schemas of these two database implementations are vastly different and DTS applies the data transformations from one database schema to the other. The DTS Workflow manager allows for the sequencing of the data extraction process. This is an essential feature to allow ensure referential integrity in the target database.

3.2 *Database*

MS Visio was used as a database design tool. The design was then exported to SQLServer when the design was completed. This process allows for much more rapid development time. The database implementation uses TSQL triggers and stored procedures to ensure referential integrity.

3.3 *XML Server*

The FdmsXML server hides the details of the implementation of the database tables. The API provided by the XML Server consists of requests written in XML and transmitted over the HTTP protocol. The XML Server listens for HTTP POST requests on a given port. When a request is received, it interprets the request, queries the database and returns the results through HTTP formatted as XML.

The FdmsXML Server runs on Microsoft Windows 2000 inside the new .NET environment. Written in the C# (C Sharp) programming language, the server leverages much of pre-defined functionality provided by this environment for database access, XML manipulation and HTTP communication.

The XML Server has some advantages over the alternative of using Java Database Connectivity (JDBC) directly by client applications. First, it insulates client applications from minor changes in the database design. Secondly, SQL Server provides some native XML capabilities that could also be leveraged by the server to improve performance and efficiency.

3.4 *Client Application*

The client website provides the user with functionality to gain access to the site from which the financial summaries or reports can then be selected.

The dynamic portion of the website consists of JSP pages running inside the Tomcat Module for Apache on a

Sun Sparc workstation running Solaris 8 operating system. The JSP pages forward client requests to the XML Server. Server responses are efficiently converted from XML into HTML through the use of XSLT transformations. Much of the site's logic resides in the XSLT source files. The Apache Jakarta Xalan library provides the XSLT transformation engine for the transformations.

User information is saved in a separate SQLServer database.

4. CONCLUSION

The SuperFDMS provides NOAA Research headquarters management with secure, near real-time, web-based access to financial information consolidated across the eighteen NOAA Research laboratories and program offices throughout the country. SQLServer has proven to be a good choice with plenty of development resources and tools to help smooth the development process. Valuable lessons were learned from this development and some components of this architecture will without doubt be incorporated in the new version of the FDMS application.

5. ACKNOWLEDGEMENT

This publication is (partially) funded by the Joint Institute for the Study of the Atmosphere and Ocean (JISAO) under NOAA Cooperative Agreement No. NA17RJ1232, Contribution #877 and PMEL contribution #2418. The views expressed herein are those of the authors and not necessarily those of NOAA or any of its sub agencies.

6. REFERENCES

J. F. Fabritz and N. N. Soreide, 2002: using Java, XML and XSLT to create secure internal access to local NOAA Research science project and budget information. 18th International Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology AMS, 13–17 January 2002, Orlando, FL (submitted).