MADILIS, the Microsoft Access-based Digital Library System

Scott Herrington, Ph.D.

Philip Konomos

Arizona State University Libraries
Herrington@asu.edu
Konomos@asu.edu

Abstract. The ASU Libraries' staff had considerable experience creating digital library systems to satisfy the needs of a major university library. These systems were designed to be high performance, large scale systems, capable of supporting very large, multimedia databases, accessible to large numbers of simultaneous users. Using this experience, the staff set out to design a digital library system that could satisfy the needs of small libraries. A small digital library system cannot be simply a scaled back version of a large system. The primary factors driving the design of a small system are cost, scalability and technical support. The resulting digital library system, named MADILIS, is designed to satisfy all of the criteria for a fully functional digital library system, while also meeting the cost, scalability and technical support needs of small libraries.

1 Introduction

The ASU Libraries' created a web-based digital library system for its locally created citation databases in 1997. This digital library system utilized a UNIX server and an unlimited user software license for BRS/Search. BRS/Search is a full-text database and information retrieval system. BRS uses a fully-inverted indexing system to store, locate, and retrieve unstructured data. The system is capable of providing rapid query response time for databases containing up to 100 million documents, each with up to 65,000 paragraphs.

The digital library system developed by the ASU Libraries included a web based search interface *and* a web-based maintenance interface, and supported links to a full range of multimedia objects. When this system was placed into full production, library systems' staff began receiving requests from other libraries for more information. Many of these requests came from small libraries without the staff to implement their own versions of this system, even when the ASU Libraries was willing to share its digital library architecture and source code.

2 Digital Library Criteria

Using a small grant from the ASU Libraries, a project was initiated to develop a digital library system capable of satisfying the needs of small and medium-sized libraries. Based on direct experience, numerous requests from other libraries and a review of library literature, the following criteria were established for this system:

1. *Low cost*, including all hardware and software components;
2. *Technically simple* to install and manage;
3. *User Friendly*;
4. *Multi-user* (including both searching and maintenance);
5. *Multimedia* digital object enabled; and
6. *Platform independent* (including both client and server components).

The project team was in unanimous agreement that every effort would be made to ensure that each of these criteria was satisfied in the design of this small-scale digital library system.

3 Design Considerations

Some design points were immediately obvious to the project team as it sought to create a system that would meet the established criteria. The client would have to be a web browser, to ensure low cost, user friendliness and platform independence. Using a standard web browser as the client also provided a solution to the issue of multi-media support: both of the major web browsers available can be used to view standard images, as well as document formats, either directly or through the installation of free plug-ins. In addition, both Windows and Macintosh operating systems include software that plays sound and video clips; these applications can be launched from standard web browsers. While this solution to supporting multi-media restricts somewhat the choice of media formats, it still leaves libraries with viable options that encompass the standard formats of choice for most library multi-media materials.

The choice of hardware to support the digital library system was based on two factors: cost and familiarity. When both of these factors were weighed, the obvious decision was to use a PC workstation as the server platform. PC workstations are the most common computers in libraries today, and—depending on configuration—are relatively affordable for most libraries. These workstations are also fairly easy to install and maintain, making technical support less of a problem.

Another fairly straightforward choice was the selection of PERL as the programming language for this project. PERL is free, readily available, platform independent and easy to learn—especially for programmers with experience in the C programming language. There is also a wealth of helpful PERL code and code modules available for download from many Internet sites.

Three critical challenges faced the designers of this new system, challenges that had not been faced in designing large scale digital library systems. These challenges were based on the need for *low cost*, *scalability* and *technical simplicity* in the new digital library system. The solutions to the problems created by these challenges were often interrelated, and these issues were a fundamental consideration in all design decisions.

3.1 Low Cost

BRS/Search was the designers' first choice as a database management system for the small-scale digital library system, but cost led to it being ruled out as a viable option. Even considering all of the very desirable features BRS/Search provides, a multi-user license for the character-based system running on a Microsoft Windows system cost several thousand dollars. A single-user license for BRS/Search cost almost $1,000, and would not support multiple users without considerable programming.

The search began for an alternative database management system. The designers knew that they needed a system that was low cost, scalable, multi-user, and robust. The most logical choice was the Microsoft Access database management system. This relational database management system (RDBMS) can be purchased at very low cost and is a fully functional system with a Window's client interface. It runs on inexpensive PC workstations, under both Windows 95/98 and the Windows NT operating systems. It is also a surprisingly "open" system: programs can be written that provide secure connections to Access databases created by the Windows client. Once opened, the tables in an Access database can be searched and entries modified using standard SQL commands. The Access system is inherently multi-user, and has excellent security features.

Using a relational database management system for bibliographic data does create some problems for the designers. Whereas the MARC record is designed to accommodate varying repetitions of fields from one record to the next, the RDBMS does not easily handle this concept. Creating additional fields in one record creates the same number of additional—and blank—fields in all the records in a database table, along with all the overhead this brings. This problem is solved through programming, by creating additional tables--for repeating fields--linked to, but separate from the main record in the primary table.

3.2 Scalability

Operating system upgrades often improve performance, but do not normally solve serious scalability problems. While the database management vendor also provides software upgrades, these too cannot be relied upon to solve serious performance issues. Another way to scale up a system is to move the database(s) to a more robust, higher performing database management system. Fortunately, Microsoft provides a direct path for the migration of Access databases to its enterprise level RDBMS, Microsoft SQL Server. Only minor modifications have to be made to the digital library system to accommodate this change, since the same programs—based on communicating with the RDBMS using SQL commands--will work with either RDBMS.

3.3 Technical Simplicity

Designing a system that is easy for technically naïve people to install and maintain is not a simple task. Computer software and hardware are not inherently simple to manage. Every decision made during the design of MADILIS took into consideration the need for minimal, straightforward and simple maintenance. Some of the most obvious and significant choices that affected technical simplicity included:

1. *Client software*: the system is designed to allow patrons to use any of the standard—and heavily used—web browsers that are freely available;
2. *Server hardware*: while all hardware can be difficult to support, choosing a PC workstation provides a platform that can be most easily supported by small libraries;
3. O*perating system*: finding someone to support any of the Microsoft Windows operating systems should not pose a serious problem for libraries, and the release of Windows 2000 consolidates the competing Microsoft operating systems into one;
4. *Programming language*: PERL is as accessible as any programming language available; it is free, easy to learn, and portable (with the caveat noted above); and
5. *Database management system*: Microsoft Access is inexpensive, easy to use, comes with a friendly client interface, creates accessible tables, and runs on small PC workstations.