DATE: January 29, 2004

TO: UMS Library Directors

FROM: Marilyn Lutz

RE: Digital Objects Management System Proposal

During 2002-2003 I worked with Fogler staff to develop a prototype system to manage digital objects using finding aids in Special Collections, and in particular, the finding aid for the William S. Cohen papers. An XML based system was designed to allow search and retrieval at all levels of an EAD encoded finding aid (technical detail below).

The idea behind the development of this prototype system was to evolve it into a larger system or repository to manage metadata and associated digital objects in all formats from the collections of the UMS Libraries.

The UMS Libraries are and will be purchasing electronic books and other digital objects, and converting documents to digital full text, audio or video from their collections or those of other campus units. The digital object management system will provide the libraries with infrastructure to store these digital objects, manage access, and integrate the metadata across collections. Here are a few of the key ways in which the system can support broadening access for the campus libraries.

1. **Combined access to Special Collections: FK – Acadian Archive, USM – Special Collections, Oscher Map, L/A – Franco American Heritage, UM – Special Collections, Cohen Archive**
    - Access to a special collections repository of metadata for all collections through a central interface with links to digitized material as it becomes available

1. **Technology to support e-book collection development: storage and access to electronic books for each campus**
    - System to accommodate multiple formats of e-books
    - Full text indexing of e-books
    - Central database of e-books with links from URSUS

2. **Creation of a metadata repository for gateway access to local databases and other databases OAI compliant**
    - Technology to support an open archives initiative (OAI) service provider and harvester protocol
    - Ability to search all local resources with Encompass
    - Like Special Collections, a single interface to an index of multiple collections, including e-journals, with links to digital resources

3. **Technology to support resource development in digital formats, including database structure, full text indexing, storage, access interface**
    - Digitization of local library and campus resources: Theses & Dissertations, Listening Center Online, Art Gallery, Hudson Museum, etc.
    - Digital Reprint services for other campus units

4. **Authenticated user access through a single interface**

5. **Expansion of resources available to distance education students**

There are significant cost savings in taking advantage of the system development work already completed for the Fogler prototype rather than purchasing or building a new system from scratch. This is a request for funding from the bond budget to adapt and evolve a digital object management system for use by the UMS Libraries, using system development work already completed. Outstanding work includes:

1. Expand database to include –document type data so that electronic documents in proprietary formats or even yet-unknown formats can be displayed with usable links

2. Produce a set of generic access control routines with a single, universal interface adapted from existing, hard coded scripts for specific databases (Indexes and Databases, Theses and Dissertations)

3. Map current collections to the existing digital object database and build the metadata harvester interface and mappings.

4. Design front-end requirements and produce templates; integrate into server scripts for dynamic content delivery.

Funding Request: $12,000

Technical Detail – Fogler Prototype

The system structure uses an SQL Server database that is designed to capture all of the searchable information from EAD encoded finding aids. The SQL Server database contains several sets of tables that correspond almost directly to certain elements of the EAD DTD, and a table that contains data to be indexed by the full text search engine. A finding aid load process maps data between the SQL Server Database and the finding aids using two XSL Schema files. The raw finding aids are not loaded directly into the database, but are first transformed by a series of XSL transformation programs into a format that corresponds directly to the database structure.

To assist with digital object management the entire Fogler web has been configured for full text indexing on the web server. A PDF indexing filter was added to the indexing service to allow full text searches of PDF documents. Documents can be linked into the object database via filenames or URLs, thereby allowing full text searches in the file system to obtain metadata from the database.

The key to management of digital objects is identification and storage of common metadata elements. The DOM is designed to store metadata harvested from other databases. The centralization of metadata storage solves a number of the problems of unified searching and, as a central repository, can be used for other purposes such as universal resource names (URN) and /or digital object identifier (DOI) resolution; centralized access rights administration, validation and control; and an access point for providing metadata to other metadata harvesters such as the Open Archives Initiative (OAI

Digital Object Management System Implementation Plan
C.Meadow 2/18/2004

We propose to take the existing Special Collections Digital Objects database and generalize it to a system-wide database for digital object management. The DOM DB currently is a nearly complete system that was started as an extension of the EAD finding aids and was later adapted to other special collections.

Task List for Converting Special Collections Objects Database to DOM Database

I. Systems Analysis

In the systems analysis phase we will review overall systems requirements needed to generalize existing work to a system-wide DOM database. The product of this phase will be a database design that will be applied to the current system to prepare it for the implementation phase.

1. Analyze existing and proposed collections of digital objects to identify additional metadata elements that need to be added to the DOM DB. We expect that relatively few elements will need to be added because the current DOM structure is adequate for nearly all special collections. **(8 hours)**

2. Generalize existing structures in the DOM database. Many of the child tables that store multivalued attributes were based on the Encoded Archival Description (EAD) structure used for finding aids. Finding Aids are only one of many collections, and even a cursory analysis of other collections suggests that EAD-based child tables can be generalized to handle multivalued attributes in other collections. **(4 hours)**

3. Analyze administrative requirements and metadata elements needed to store them. Most of the administrative requirements center around user authentication and access restrictions. A system-wide IP and barcode authentication structure is already in place that can identify a user's campus affiliation and IP address. Other possible restrictions include date restrictions, copyright restrictions and course enrollments. Structures needed to store restrictions are not yet fully in place. **(12 hours)**

4. Analyze the requirements of the Open Archives Initiative metadata harvester and ensure that the DOM database is suitably structured to receive output from the harvester. This analysis is expected to produce structures called "metadata maps" that will be used to map collection to XML inputs to the OAI Harvester. The metadata maps will serve not only to map collection data to the harvester but will also map descriptive collection metadata to the DOM DB **(8 hours)**

5. Revisit existing full text query parsing code and design a consistent across the board query language based on an agreed-upon list of rules. Searches in the DOM DB will almost certainly be conducted primarily using SQL Server full text

searches, often in conjunction with full-text document searches in the file system. Currently we have perhaps a half-dozen slightly different implementations of what amounts to a quick and dirty pattern-matching parser for a query language. This was initially developed under budget pressure and is not terribly robust, nor is it easily documented. A simple, general query language for all searches can be implemented by a robust, recursive-descent parser, documented once and for all, and can be reused in many places. **(Analysis and documentation 12 hours)**

Total Programmer time for Systems Analysis: 44 hours.

II. System Implementation

The implementation phase has 4 major elements.

1. Restructure database and adapt existing web scripts to new structures. Existing web scripts will be adapted for the short term only; to keep them usable by the public while project implementation proceeds. **(10 hours)**

2. Build structures and programs needed for metadata harvests.Programmer will be responsible for interfacing existing collections to produce suitable inputs for the harvester. **(20 hours)**

3. Update existing front end to a solid, fully functional front end for manual data entry and editing. **(15 hours)**

4. Implement restrictions as ASP server-side includes or SQL Server stored procedures and link DOM to user authentication system. **(20 hours)**

5. Implement reusable recursive-descent query parser. **(20 hours)**

6. Recode existing web searches to use new structures and systems and implement new searches for additional collections now defined in DOM DB. One of the goals of this phase will be to produce either parameterized, reusable search scripts or else search templates that can be adapted to other collections with minimal programming effort. **(35 hours)**

Total Programmer Time for System Implementation: 120 hours.

Total Budgeted Programmer Time: 164 hours @$60/hour: $9840