

VIUS Report 8.8

Case Study in Microcomputer Support for Individual Collections

The decision to investigate a microcomputer-based database approach to image collection management was based upon several factors. First, some focus group participants who had already accumulated a number of digital images wanted assistance in organizing their collections. Secondly, it seemed advisable to anticipate the needs of those who might be building or refining an individual or small project collection with the aim of its eventual inclusion in a public or shared server-based collection. Finally, the effort was opportunity driven. A member of the faculty (“Prof. T”) was actively engaged in building a collection of images for a new course, recognized the desirability of coordinating the approach to scanning and metadata so that these resources might, in the convenient future, become part of a shared resource. LionShare’s features, when fully developed, may have met Prof. T’s needs, but since that development was incomplete we ventured into a more modest effort to provide support. Since Prof. T had a graduate student building the collection, would it help to share our cataloguing procedures? Since Prof. T was typical in having modest resources to draw upon, would writing an application in a readily available application (like MS Access) provide useful support?

The written procedure used to train and guide the graduate student catalogers of our image database service was sent to Prof T’s graduate student, who mimicked the fields of our CONTENTdm™ database using spreadsheet software.

A VIUS project team member devised an MS Access database to serve as a microcomputer-based image metadata collection platform. The database had a simple relational structure. Field lengths were ample, in anticipation of broadly defined, possibly narrative entries. The database included an input form to simplify data entry, as well as the means to include a pre-built lists of values in lookup tables for specific fields, enabling the creation and use of a pre-coordinated controlled vocabulary in those fields. Record sets could be exported in a delimited format or in XML. Data from the existing spreadsheet was imported into the database and a set of controlled values were set up for selected data fields (i.e. ‘Subject’).

Discussions were held to gather user impressions of the database, and some of the results were, at first, surprising. Although the users appreciated the form-based data entry, the sense of their reaction was that the database was not a large step forward from their experience using the spreadsheet. While the pre-built values attracted comments such as “nice”, and “handy”, they saw the utility of such value lists in terms of the keystrokes saved, rather than for their role in encouraging data consistency.

Although from the start the database had been portrayed as a data-gathering, data entry, record -building tool, the users expressed a desire for features of the sort one might expect from a more typical commercial image management tool targeted at a home user. They wanted thumbnails, for example, as well as a means to select records for export with a check box, and automated exportation into PowerPoint.

The impression left from these discussions was of a difference in definition of the term “collection management” between those involved in implementing the server-based system (such as our CONTENT dm application) and those interested in “managing” a personal collection. It seems clear that data architecture issues that are self-evident in the implementation of a large-scale system cannot be assumed to matter to users interested in managing a personal collection. To the keepers of a server-based set of collections, data integrity is paramount. The individual collector might very well say that thumbnail displays and light table functions matter most.

One might possibly accommodate these disparate sets of interests by designing a personal collection tool with a feature set rich enough to make its use attractive to individual collection builders, but designed to maximize the creation of robust records that are “as ready as possible” for import into a server-based system. This suggests that automated data entry, to the extent that it can be built into such a system, ought to be. The pre-built values lists such as those built into the trial database are one example. Beyond this, technical metadata might be gathered from a file’s MIME type, its size, date of creation/modification, etc. User information gathered once might be automatically supplied into descriptive fields, at least as editable defaults.

There is much to be said for encouraging the creation, even by individual users, of record sets with stable, known characteristics. On the other hand, it seems unlikely that it would ever be cost-effective to develop an in-house product as feature-rich as even a modestly-priced commercial product. Perhaps one could, on the institutional level, settle on a small set of “recommended” commercial products for individual or group projects, and attack the data integrity issues through best practices guides and migrational aids.

Ultimately, regardless of the sources from which they gather them, the responsibility for data in large, curated collections will always fall to the builders and keepers of those large collections. Should Prof. T’s collection ever be added to our CONTENTdm™ image database service, the process should be facilitated by the similarities in the scanning specifications, the data fields, and some of the data values – insofar as these can be maintained by Prof T and his assistants while using only basic tools. Perhaps the lesson for the development of LionShare is that thumbnails, light table features, and automatic assignment of metadata should be given priority where feasible.

Michael Pelikan and Henry Pisciotta
10/2003