currently because we are so bombarded with imagery every waking moment. It is important to become visually literate, or visually discerning, in order to survive.

There was an interesting Twitter (http://twitter.com/) about your presentation at the SLA conference that said: "Tamara Moats rocks. She led us through an examination of Homer's *The Lifeline*. We all *do* have instincts to interpret what we see." Do you feel that you are, as they say, "changing lives?"

I am not so arrogant as to think I can change lives, but as a museum educator, I was trying to make the world a better place, and to help create a museum that stimulated thought as well as gave pleasure. As a teacher, I hope I have motivated a number of students to look at the world differently, and a few have decided to major in art history!

Thanks very much Tamara, I enjoyed both hearing your engaging presentation as well as chatting with you!

Participants at Tamara's Standing Room Only Special Libraries Association Workshop on Visual Thinking. Seattle Conference, June 2008

Tamara Moats can be reached at: Tamara.Moats@bush.edu

Review of the Cuadra *STAR* Application for Library Automation - Managing both Traditional and Electronic Library Collections

This is a review of the Cuadra STAR application which is used for an article level database in an historical digital collection. The review focuses on the creation of the database, input and output fields and webpage creation. The review also includes the step-by-step process used to create the collection and the pros and cons of each step. Overall, the experience with building our database and web interface in *STAR* was extremely positive. If one has experience with database creation and HTML, this software is ideal to use, however, *STAR* would benefit from some work on their user interfaces.

Digital Literacy

About Cuadra *STAR*
Cuadra offers a comprehensive suite of customizable, task-oriented knowledge management solutions. Each Cuadra solution is designed to meet the professional standards in its field, as well as provide for fast, precise web-based retrieval, to enable users to find the information assets they need, when they need them.
- Collections Management: for managing the full range of archive, museum, photographic, and image collections
- Library Automation: for managing both traditional and electronic library collections
- Knowledge Management: for managing the "corporate memory"—the vital information being created and acquired by your organization
- Media Management: for managing all kinds of visual collections, including those

used in presentations
- Records Management: for managing both electronic and paper records
- Vocabulary Control: for creating and maintaining thesauri to facilitate the indexing and retrieval of your data

Background
The Digital and Multimedia Center at my organization has a collection of historical items across many different subjects. Our collections are currently housed in a *mySQL* database that was created in-house and served through a *JSP* coded front end. Our current functionality has browsing for individual items at the book level and searches only at the title and author level. We have several serials digitized in our collection but no way to offer article level searching of our serials. We decided to create a prototype of article level searching to see if the software would work for us and how difficult it would be to create and maintain.

This review focuses on the creation of the database, data entry and display of the database information, including a walkthrough of each step of the process. Since we already had an installation of the *STAR* database I did not install or set up any of the software on the server itself so I will not be reviewing that portion of the software.

Metadata Setup
One of the advantages of working with *STAR* is the ability to create whatever metadata schema you'd like to use, although it has a few constraints. It is also versatile enough to allow you to develop the schema you need. I worked with our metadata librarian to choose a schema for the prototype. We wanted to make sure that *STAR* could accept the fields we would like to use and make sure that it was forward compatible with any other system we would migrate to in the future. We chose *Metadata Object Description Schema* (MODS) to start. I worked with both the metadata librarian and the database

administrator of STAR to negotiate what would work within *STAR*.

For the most part, the input fields of MODS would work in the *STAR* database, however, we did have to do some modification of the schema to make it work. The MODS schema has subfields of subfields, however, *STAR* would only accept one subfield. For example, for the tag <subject authority="naf"> which contains a subfield <name type="corporate"> which has subfield called <namePart>. Because STAR does not support subfields with subfields we needed to create a work around. We decided to include <subject authority> manually in the data dump if we move to a new system. Since the material we were putting into the database was relatively homogeneous, this seemed to be a good work around.

Overall, choosing the metadata schema and matching it to what *STAR* could do was a relatively painless process. That it was flexible enough to handle the different metadata schemas is a real positive. We could use different metadata schemas for different databases and collections which makes the database system extensible and desirable.

Software Setup

In order to access the *STAR* database to create input/output fields and to enter data into the database, MSU utilizes what Cuadra refers to as STAR Classic. I use a secure shell program with VT200 emulation. I used *Hummingbird* software because my institution has a site license for it. I have used secure shell programs in the past so I felt comfortable with the interface. It's important to note that Cuadra does offer web based modules for the database creation and data entry. We chose not to use the modules at my institution so our implementation is different than some of the current *STAR* users. I did not use or evaluate the web based modules and it would probably be worthwhile to ask for a current demonstration of any software you are thinking of purchasing.
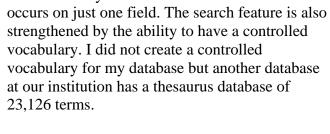
The other module we use to serve the collections is *STAR Web Designer*. We use *STAR Web Designer* in coordination with *Dreamweaver* to build the web interface to the database.

Input Field Setup

Once past the interface issues, setting up the input fields was relatively easy. It took me far longer to decide what fields I wanted than to actually set it up. The first step was to create the database we will be working in. One of the positive aspects about *STAR* is the ability to host multiple databases, with different schema. Once the database is created, the user is offered a list of options which are fairly straightforward and easy to understand. One of the downfalls of the database is the character limitation for input field labels. While the output field labels do not have a length limit, the input fields labels allow for a maximum of 16 characters. I do not understand the point of the limitation and it can make for challenging naming of the fields. On the bright side, the output fields are not limited at all. The process to set up the input fields was quick and easy.

Search Fields

You can create as many search possibilities as you need in *STAR*. Since my database was fairly simple, I created four different search types. You can combine

fields for a keyword search or have a search that occurs on just one field. The search feature is also strengthened by the ability to have a controlled vocabulary. I did not create a controlled vocabulary for my database but another database at our institution has a thesaurus database of 23,126 terms.

Output Field Setup

One of the features I loved about *STAR* was the output field setup. The output fields were just as simple to set up as the input fields, however, when creating them think how the webpage record should look. You can combine the input fields into one field to create an output field which makes designing the web interface much easier. For example, the metadata schema we use calls for file size and page numbers. I created an output field to combine those two to shorten the record and group the information together. I loved the versatility of this feature. Another feature I liked was the ability to track date of creation as well as the date the record was revised. These fields are system generated and the user does not need to edit them.

Web interface

The STAR Web Designer component was, for me, the most difficult to understand and use. I do have a background using databases all the way back to 1990 so I probably felt more comfortable with the back-end database component than many readers might. However, I have been creating webpages since 1995 so I am not a novice web developer. What I found difficult to understand were some of the terms they used to create the webpages. There are Variables, Report Definitions and Pages components that are used to build the web interface. It took me several days and many attempts to really understand what each of those components do. Once I understood the terms and how to use them, building the pages became very easy. The process of trial and error, rather than reading the manual, has always been the best way for me to learn. I recommend that you use whatever way works best for you.

The second difficulty I faced with STAR Web Designer was the interface itself. The page you work on is cluttered with boxes, sometimes with redundant information. The use of screen space is not intuitive and you have to scroll left and right to view the information in the columns. This was a minor annoyance but also a confusing one.

Best Features of *STAR*
What I absolutely loved about *STAR* was the ability to create complex HTML pages with Dreamweaver without knowing how to program. For me, this was the strongest and best feature of *STAR*. Using Dreamweaver I was able to put in simple STAR codes that resembled HTML to create a website. With our *mySQL* database, I would have to learn how to code in order to accomplish the same outcome. This process is a little more difficult if you have not coded HTML or used *Dreamweaver* in the past, however, it is not as difficult as learning a programming language. In the example below, you will see code for *STAR* identified as "STARweb_type." The code shown is to create the "Records/Screen" display on the search results page. The other feature I loved about *STAR* was the flexibility of the display. You can display in HTML however you want to display the information. The design is truly what you make it. If you visit their different client websites you cannot really tell that they were built on the same platform.

When I was evaluating what program and database to use to create our article level material I talked with our programmer and our web designer to find out how long it would take them to build the website to our specifications. They quoted me a time of approximately 80 work hours. I was able to build the database and create the web interface in approximately 60 work hours, but that time also included determining schema, output field combinations, and actually learning how to use the software. The next site I build will not require as much time. I also liked the ability to create it myself, without having to rely on the scheduling of programmer time, designer time, and issues of communication.

I did have great assistance from the TIC database administrator who patiently answered my questions about the system while I was learning. I would have been able to create the database and web interface without him but I would probably have had to rely more on the manual.

About Cuadra and Documentation
Information about Cuadra and *STAR* can be found at http://www.cuadra.com/. I enjoyed looking at their different customer sites to see how they were built and designed, and because of the wide array of information located there. Their clients include National Archives and Records Administration, AARP, and The Vietnam Archive to name a few. *STAR's* strength also includes its extensibility for larger collections. While the collection I was creating was a prototype and, even when fully populated, would be around 15,000 records; they have other customers with much larger datasets. The Virginia Historical Society has over 150,000 items. Our Turfgrass Information Center has over 130,000 records. It is useable and scalable for small datasets to large datasets. As I mentioned in this article, I did not look deeply at the documentation. I am much more of a hands-on learner. I did look at the documentation but it appears to be more about showing examples. While this type of documentation can be helpful, sometimes more descriptive information is needed.

Elizabeth J. Bollinger is systems librarian at the Digital & Multimedia Center/Vincent Voice Library,
Michigan State University.
bolling7@mail.lib.msu.edu