Potentials of Dynamic Database-Driven Web Sites

Francie C. Davis and Laura Pope Robbins

Introduction
With more and more resources available through the Internet, the process of providing quick, easy, accurate access to these resources can be a daunting, overwhelming, time-consuming prospect. URLs change without notice, database providers add and delete journal titles, and our customers expect service 24/7. Database-driven web sites address this problem as they simplify access to the resources provided by your library. They are easy to maintain and update, provide an efficient means to eliminate errors over multiple web pages, as well as, a means for fast, efficient searching.

With that in mind, the librarians at Dowling College have developed the Journal Locator and Database Locator. The Journal Locator provides a dynamic listing of every access point we have for a journal title, whether through an aggregator database or physically within the library. Our customers can then determine where the title can be located in full text and link directly to that database. The Database Locator provides the same type of subject area dynamic linking. Both locators are dynamic databases, which provide accurate and up-to-date one-stop-searching for site visitors while simplifying and eliminating the redundancy in web page maintenance. They provide dynamic listings, create an interactive resource that can be manipulated, are fast and accurate, and are time savers that either eliminate or facilitate time-consuming tasks. This paper will address the advantages of database-driven web pages, discuss important design principles, and point out the critical success factors for the creation of such sites.

Discussion
In an environment where electronic resources are changing day-by-day, minute-by-minute, providing access to them is an ever-increasing challenge. The known interface of the morning is gone by the afternoon's instruction session. Aggregators add and drop full-text journal titles. Names change with the stock market. URLs change overnight and must be...
updated for our home page web links. A goodly portion of a library's budget is spent on the illusory electronic resource, but how can one feel confident that those resources are being maximally used? Students want the quick and dirty answer and the full-text online article at home. They want a direct link from their search to the full-text article in question that they will then print at your library's expense. Given that scenario, the question is, How can we ensure our customers accurate and efficient access to our resources? We think that the answer is dynamic database-driven web sites.

At Dowling College, we have developed two major database-driven web sites, the Journal Locator and the Database Locator. Both databases have greatly enhanced our web page and become the top pages accessed by our customers.

### Database Locator

The Database Locator was created to address the concerns itemized above. There had been a costly proliferation of databases, and we found that we although we were spending a large percentage of our budget for online resources, we were providing only five subject access points. As such, we were not helping the students to reach the information they sought. By creating the Database Locator, we expanded the subject searching from five broad subjects to more than 60 subjects. For instance, we went from one Natural Sciences and Mathematics heading to at least 13 more specific headings. The expansion in subject points allows the user many more possibilities for contact with the databases titles. While the broad Social Sciences category might not click, the more specific Psychology category might be just the ticket. (see Figure 1)

Maintenance for the hundreds of database titles was a time-consuming nightmare. One URL change for one database on just the seven pages (five subjects, one annotated, one alphabetical) we were supporting, could take an hour or so per week and was an invitation for errors. And that was only one change. Since there is no efficient way to do a global update on static HTML pages, each change required the physical handling of each page on which the link resided. Multiply that by all the actual changes that are required and you get an idea of the labor involved. By switching to a database-driven web page, that one change takes less than 30 seconds for ALL the web pages (more than 60 subject, more than 60 annotated, an alphabetical with and one without annotations) that are now maintained. Putting our databases into a centralized resource made it possible to make global changes to our website, since the update is made only to the one place. Making a total of one change instead of individual changes on each and every page, guarantees the accuracy of the pages. That's quite a difference in effort and quality control.

The variety of access points we're now able to provide more accurately meet the needs of our students. We're giving them dynamic content. The resources are delivered only as requested and are more specific to their needs than static pages could ever be. Because the students are being presented with more specific subject headings, they are not being overwhelmed with a longlist of databases to wade through. They can choose from annotated or non-annotated lists, and these pages are easily integrated across the web pages through embedded links.

### Journal Locator

The impetus for creating the Journal Locator was a...
potentials of dynamic database-driven web sites

April 10–13, 2003, Charlotte, North Carolina

A severe budget crunch precipitated a long hard look at our periodical subscriptions. We determined that we would come out ahead if we cut our paper subscriptions, but subscribed to new full-text aggregators. By doing so, we could pick up many of the titles we were cutting and add new ones. But the question remained as to how to inform our customers of what we had and where they could find it. To do so, we created the Journal Locator as an access point to all information about more than 29,641 journal subscriptions that are available directly from our web page and our OPAC. Using the Journal Locator, a student can search on a journal and find out what type of access we have and what our holdings are for that title. If the journal is available in a full-text aggregator, there is a link directly to that aggregator. (see Figure 2)

We chose to create our own Journal Locator rather than go with products such as JAKE (http://jake-db.org) or Serials Solutions. JAKE could not include our library’s physical holdings and the format could be difficult for students to understand. In addition, JAKE is not updated regularly, and it made searching for a journal a two-step process. Students must first locate the journal in JAKE, and then go back to the database list to find the appropriate aggregator. We did not use Serials Solutions because it did not exist when we started the project. After it did become available, we stayed with the Journal Locator because Serials Solution did not, at that time, provide a link from the OPAC or the ability to search for a specific journal. These were capabilities that we felt were essential and were already components of our database.

The Journal Locator currently provides access to all of the titles located in 16 of the aggregator databases to which we subscribe. In addition, it includes information on our physical holdings and to selected e-journals that we receive either in conjunction with a physical subscription or that are freely available over the Internet. We’ve found that there are several advantages to creating our own Journal Locator. We can customize our search results and change the search options easily to better meet the needs of our students. For example, we now offer our students Boolean searching. It’s very common for aggregators to enter titles differently, especially if there is a subtitle. This can cause problems if all that is allowed is a full title search because access points will be missed. With the option for Boolean searching, searching on subtitles is no longer a problem. (see Figure 3)

Aggregators are famous for making changes monthly. With a print serials listing, this is an insurmountable problem. With a database-backed listing, this is simply maintained. New aggregators are easily assimilated into the database as well. Thus, as new titles become available to your customers, access remains seamless. This type of database also improves Interlibrary Loan services. Very often, our customers seek to Interlibrary Loan materials to which our library has access in a database other than the one they are searching. Unfortunately, they don’t know that they do. With this database, they are fully aware of every entry point for a particular title. By integrating the Journal Locator into our OPAC, we’ve provided current indexing information within our full-text aggregators for our customers as well.

Programs
It takes three components to create a dynamic-data-
base driven web site:

- webserver
- database
- middleware

In our case, we work on a Microsoft dedicated campus. Therefore, we used the Microsoft Internet Information Server (IIS) for our webserver; Microsoft Access for our database; and Active Server Pages (ASP) as our middleware.

- Access provides the raw data for the web pages.
- As the middleware, Active Server Pages are what allow the webserver and database to talk to each other. It basically provides the interface between the two by incorporating static HTML with dynamic content. This is done through the use of server-side scripting. (see Figure 4)
- ASP pages are easy to write and maintain and work well with any browser. Unfortunately, this technology is specific to Microsoft as it is integrated into the Microsoft IIS. To use it with other web servers would require the purchase of additional software, like Sun ONE Active Server Pages formerly Chili!Soft or iA SP.

Tips

- When working with any database, it is important to pre-plan what you will be doing. Before you ever put fingers to keyboard, you need to think about what your current needs are and what your future needs might be and then include this information in your database. For instance, with the Database Locator we not only wanted database titles, URLs, abstracts, help pages and subject categories, but also, information about licensing, vendors, budget costs, passwords, administration of the account, contacts and budgeting.
- Don’t put all the information in one table. If there is a new category of information, give it its own table. For instance, each of the above mentioned categories for the Database Locator has its own table. In addition, standardized data should get its own table.
- Consider what types of data will be included in the database before you begin inputting it. As with any type or project, it is easier to alter in the design phase than to retrofit. For example, can an abstract fit within a text field limited to 255 characters or does it require a memo field?
- In text data fields, you want the space available to be large enough to hold the necessary data but not so large as to cut down on speed. Note that if the field is made too small, it will cut off excess information. That is especially important to consider when you are dealing with URLs. A truncated URL won’t work.
- If you plan to use hyperlinks and put the database on the web, the field for your URLs should be defined as text NOT htm.
- While you are in the pre-planning process, be sure to ask for input from others. What do they see as the function or goal of the database? Do they have any needs for the database? All of this planning will only make the database better and more functional. The database can be fine-tuned after it is created, but
the more thought that goes into the creation, the fewer headaches you will have later.

- Before you input data and once the tables are created, you will need to establish referential integrity. What that means is that you are going to set up this relational database so that a correction made in one table will be made for you in all related tables. If you wait until you have entered data to do this, you will be making more work for yourself. For instance, the database sees an ampersand (Baker & Taylor) as different data from and (Baker and Taylor). If the relationships have been set up first, that error will be caught. If not, you might waste a lot of time trying to understand just why the relationship won’t work. (see Figure 5)

Next you will want to create a form to input all of the data. While you can certainly update the information in the table format, it will prove to be much more cumbersome. One row of information may require scrolling many screens to the right or left. A form, however, will have all pertinent information on one screen, thus, making updating information much speedier and easier. The Combo box feature facilitates the form creation as it retrieves data from a pre-selected list of data in another table. This will not only simplify data entry but also reduce the chance of mismatched data or typographic errors.

- Use Excel to standardize the varied data for the journal titles. It also allows you to split data across columns, which Access does not. When you must have repetitive data in a table, Excel also lets you copy data from one cell into many others, while Access does not.

- To query multiple tables in Access, you’ll need to create a Union Query. The one requirement of a Union Query is that the field names in all of the tables queried must be the same. If they are not, you can use an expression to rename the fields.

- Use ASP or another type of middleware to create the interface between the web search form and the database once it is on the server.

- If you are using Access as your database, it can create a basic ASP page for you. First create all of the queries you want someone to be able to do on your database. Then export them as ASP files. Once you do this, the database will create both a basic ASP page plus an HTML query form. The ASP page will contain the SQL query that you need to search the database over the web.

- Truncation symbols differ between Access and the version of SQL that is on the server. Therefore, the truncation symbol in the ASP page must be replaced with the % symbol.

**Conclusion**

Dynamic database-driven web pages are user friendly, easy to navigate, provide a one-stop shop, and are a dream to maintain. Just remember to plan before you begin.

**Bibliography**