*January 2003*

# Knowledge Management Toolkit: Options and Trade-offs

## Introduction

In spite of the progress we've made over the last decade in connecting people and documents, we still have a long way to go. Readers spend too much time and effort finding, evaluating, and tracking. Authors, teachers, and other content creators are still forced to work in a segmented environment where print, Web, and database genres co-exist but don't mesh well. The upshot is Web publications that lack editorial oversight, print publications that lack interactivity, and databases that serve only a single application.

This article explores the role of "desktop" databases in helping content creators design a more comprehensive and efficient reader experience. First we evaluate the effectiveness of various authoring tools in providing answers to typical questions posed by readers. Then we highlight the pros and cons of strategies to add database capabilities to the author's toolkit. Finally, we discuss our own authoring toolkit and its benefits.

We intentionally use the terms "reader" and "user" interchangeably because the vocabulary illustrates the problem. We like the phrase "information ecology" to describe the system and "knowledge base publishing" to describe the creation process, but we have no word that adequately describes the intended beneficiaries.

## Beyond documents

As professionals that manage information, we sometimes forget how frustrating it can be to use it. A printed report or Web page by itself doesn't provide a complete solution to the problem. A single document or file may answer some of the questions, but rarely all of them.

- How current is this information?
- How do I communicate with the author?
- How do I learn more about the author's qualifications and experience?
- Who owns the copyright?
- Who is allowed to view and change this document?
- How do I share this information with colleagues?
- Where do I find other documents like it?
- Who else has read it? What did they think of it?
- How can I learn more about a certain concept?
- How do I access a cited source?
- Which commercial information service indexes it?
- Is there a translation available?
- Are there previous editions of the document?
- Can I be notified when a new edition is published?

Providing a complete user experience requires a combination of technology, editorial discipline, and architecture. For example, all authoring programs attach a last-saved-date to a file, but this is of only passing interest to readers. What they really want are date-created and date-last-modified. Nearly all authoring programs (the technology) provide an automatic way to enter these dates — as long as the author remembers to do it (the discipline). However, a date embedded in a file can't be manipulated by another program — necessary for searching and sorting multiple documents by date. That facility requires a database or embedded metadata (the architecture).

The problem is that the software tools used to create information are document-focused, while a complete "publication" requires additional tools that are user-, policy-, or collection-based (see the table below).

| Tools for a Complete User Experience | | |
|---|---|---|
| **Question** | **Implementation** | **Comment** |
| How current is the information? | Metadata elements for date created & updated | Authoring programs can accommodate dates but it's up to the author to enter them. A database can require that the date be entered in a specific format (necessary for sorting by date). |
| How do I communicate with the author and other experts cited? | Link to contact information | MS Office allows e-mail communication if the author enters his or her e-mail address. A database can be configured to provide complete contact details, not only for the author, but also for other experts. |
| What are the author's qualifications and experience? | Bio, resume, references, other publications, citation history | If the publication process requires it, a bio can be included in the document text. A database can provide links to a resume and other publications by this author. Third party services, such as ISI's Web of Science, let you see how often a work was cited by other authors. |
| Who owns the copyright? | Rights management | A database is the best way to track rights, since the author is not necessarily the copyright owner. |
| Who is permitted to view and change the document? | Authentication | MS Office programs can be password protected. PDF documents can be password protected using Adobe Acrobat. But a database is necessary to track document access levels and user permissions. |
| How do I share information with colleagues? | E-mail, hyperlinks | MS Office programs and most Web browsers allow you to send pages or links via e-mail. Repository records in a database can be configured to send a citation and link via e-mail. |
| Where do I find similar or related documents? | Taxonomy | A bibliography, especially one with hyperlinks, is helpful if the author provides one. This function properly belongs at the collection, not the document level, with relationships provided by a subject taxonomy and thesaurus. |
| Who else has read it? What did they think of it? | Annotations/ Usage analysis | Individuals can add annotations to documents in MS Office and Adobe Acrobat. A database is best for compiling and summarizing usage. |
| How can I learn more about a certain concept? | Hyperlinks/ Taxonomy | Both MS Office and PDF can display active hyperlinks, but links can go "cold." You need a database to manage a permanent archive of source documents and the categories assigned to them. |
| How do I access a cited source? | Hyperlinks/ Taxonomy | See above. |
| Where is the document indexed? | Content aggregation | For cited sources not accessible on the public Web (and for printed bibliographies), you need to know which aggregators (i.e. Factiva, ProQuest) offer access to a specific journal title or report. This information is stored in a database and offered for sale by third parties. |
| Is there a translation available? | Metadata element for languages | Individual documents can contain links to translations, but a database is a better choice for storing this data. |
| Are there previous editions? | Cataloging & classification | Instead of storing documents in separate folders associated with the program that created them, we store them in a single Archive folder, filed by database record ID number. A database link retrieves them instantly. |
| How can I be notified about future changes or editions? | Subscriptions / Alerts | Our vocabulary and thesaurus is managed in the database. Terms are linked to document records. |

|  | Proprietary Program | IT Project | Your Own Design |
|---|---|---|---|
| Time | No development time, short installation time | Lengthy development cycle | Short to medium — depends on the designer's competency |
| Total cost | Low | High | Medium |
| Functionality | Fixed | Can be very good | Very good |
| Compatibility | Limited compatibility, if any | Can be added as a custom feature | Can be added as a custom feature |
| Updates | Depends on vendor | Depends on IT staff availability | Updates as often as needed |

## Databases as authoring tools

Documents and drawings are static information components; databases are dynamic. Documents convey original intellectual concepts; databases allow readers to find, compare, and manipulate information. Together, they work hand-in-glove to provide a comprehensive reader experience. Databases are too important as creative tools to remain an arcane IT specialty.

## Implementing databases

Three strategies are commonly used to add database capabilities to the author's toolkit:

1. *Proprietary programs*. These are essentially general purpose databases that have been customized for a specific application. Examples are Quicken (for bookkeeping), Compass (for managing bookmarks), TagGen (for managing metadata), ACT (for managing contacts). The underlying database is often Oracle or SQL Server.

2. *Contract with the IT department* (or an outside firm) for a custom application.

3. *Design your own application* using a "desktop" database product. For a recent review of seven products, see "Databases for all reasons."

Each option has pros and cons (see the table above). If you are fortunate enough to have a long lead time and an IT staff that understands your application at your beck and call, the IT contract is your best bet. The end product will be:

- full featured (can do what you want);
- able to handle future growth (scalable);
- secure;
- well documented; compatible with enterprise standards;
- robust (no "bugs").

Very few Knowledge Base Publishers have unlimited access to this kind of talent. The rest of us must choose between the other two options — using a collection of proprietary programs that don't mesh well or learning to create a custom database.

## The database learning curve

The time required to learn how to create custom databases depends on two things:

1. Your aptitude. It helps if you're willing to read the manual, think logically, and enjoy problem solving.

2. The database program you select. The two most popular work group databases are Filemaker Pro and Quickbase. Both provide document management templates (pre-built custom applications) out of the box.

With programs like these, you can create a simple database in a few hours. As you become more proficient, you can add more features, such as:

- modifying standard database templates for common functions (e.g. contact management, billing, inventory);
- designing new reports and data entry forms;
- creating new databases from scratch;
- adding features (e.g. web publishing, passwords and permissions);
- linking multiple databases together (e.g. linking author contact details with bibliographic data about their publications);
- automating functions and adding intelligence through the use of scripts (macros);
- integrating a database with other applications through XML, Web Services, and other technologies.

## Our publishing toolkit

The table below shows our favorite tools for eighteen common publishing tasks. Our custom database is used for ten of them. As the database has become both more powerful and easier to use, it assumes a larger role in our publishing workflow.

| Task | Tool | Comment |
|---|---|---|
| Composing print documents | Desktop publishing program | Both Quark and Pagemaker can easily output documents in PDF format for publication on the Web |
| Composing Web documents | HTML editor | We use Dreamweaver for creating Web pages only. Our database program handles dynamic Web publishing. |
| Composing slide presentations | Presentation program | If a slide show is our first intellectual product on a topic, we compose directly in Powerpoint. Otherwise, we import an outline from our desktop publishing program. |
| Composing e-mail messages | Outlook | For longer, more formal documents we compose in MS Word and send the result as an e-mail attachment. |
| Annotating documents | Adobe Acrobat | Acrobat can convert HTML pages and MS Office documents into PDF, which can then be highlighted and annotated. |
| Analyzing Web logs & financial data | Excel | For temporary and preliminary analysis we use a spreadsheet. |
| Composing personalized print documents (letters, invoices) for a group | Custom database | It's easy to replicate the look and feel of paper forms and stationery with the database — including colors, rules, graphic elements such as logos, and fonts. |
| Managing membership renewals | Custom database | Calculation fields and sophisticated searching capability allow us to identify membership renewals. |
| Summarizing Web logs & billing information | Custom database | Web logs and bookkeeping data are "massaged" in Excel, then imported into the database, where they are summarized and linked to the correct contact or document. |
| Managing contacts & e-mail messages | Custom database | Since we can send e-mail from within the database, we don't use the e-mail program to store contact information. We copy and paste the body of important messages into our database and link them to the appropriate contact |
| Managing bookmarks | Custom database | Bookmarks — Web page titles and URL's— are pasted into the database along with publication date, author, topic, and our comments. |
| Managing documents | Custom database | Instead of storing documents in separate folders associated with the program that created them, we store them in a single Archive folder, filed by database record ID number. A database link retrieves them instantly. |
| Managing organization scheme (taxonomy) | Custom database | Our vocabulary and thesaurus is managed in the database. Terms are linked to document records |
| Publishing indexes | Custom database | Selected records from thesaurus portion of the database are published on the Web as "live" indexes. |
| Managing member passwords | Custom database | Contact records for members are linked to specific passwords and billing data in the database. |
| Applying Web site security | Authentication program | Our authentication program controls access to special folders or directories on our Web site. |
| Applying document security | Adobe Acrobat | When we want to control access to specific documents (as opposed to folders), we convert them into PDF format and use Acrobat's security feature. |
| Data transfer among applications | Custom database | We use MS Excel to edit tab-delimited files before the data is imported into our custom database. We find XML too cumbersome. |

We have avoided using proprietary programs, although we have tried many of them. The overhead involved in integrating data with other programs, installing upgrades, and remembering different set of commands doesn't justify the cost. Customizing our own database to add these functions was a better investment. The database is the engine that runs our entire business. The office suite and other programs are accessories that provide specialized text processing and communications functions.

**Benefits**

The major benefit of incorporating a custom database into our content creation process is the ability to create new products and offer better service to members and clients at minimum expense. A secondary benefit is the ability to move data from one application to another with relative ease. For example, user passwords are currently stored in at least four different applications, but our password database is the final authority.

Finally, a custom database is a

kind of insurance policy against technical obsolescence and market volatility. Having our data locked up in a proprietary program could mean costly software upgrades or time-consuming and error-prone migration to a new system. With a custom database, we control our own destiny.

**Conclusion**

For years, databases have been the basic building blocks of nearly all information management systems. Now they are assuming a broader role as content creation tools — on a par with word processors, presentation software, and page layout programs. Powerful and easy to use, desktop databases can be customized by nonprogrammers to add a wide variety of dynamic functions that give readers a multi-dimensional experience well beyond what they can get from documents and files alone.

---