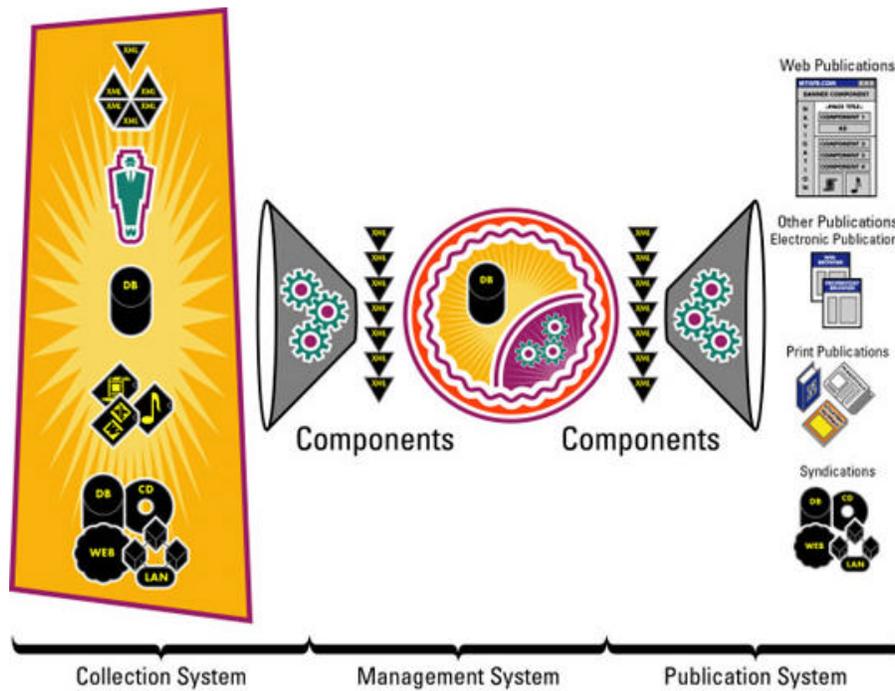


Understanding Content Management

A CM Domain White Paper

By Bob Boiko



This white paper is produced from the Content Management Domain which features the full text of the book "Content Management Bible," by Bob Boiko. Owners of the book may access the CM Domain at www.metatorial.com.

Table of Contents

Table of Contents	2
Defining Content Management	2
The Content Management Industry	3
Systems for Managing Content	4
The nominal Web CMS	4
Page templates	5
Status functions	6
Outlining and organizing functions	7
Deployment tools	8
The dynamic Web site	8
The full Web CMS	10
The Enterprise CMS	11
Summary	12

I assume that most people come to this white paper because they want to know how to make large and well-managed Web sites. You can learn that here. In the process, I hope that you also find that content management isn't about Web sites, although that's where it's mostly practiced today.

Content management is about gaining control over the creation and distribution of information and functionality. It's about knowing what value you have to offer, who wants what parts of that value, and how they want you to deliver it. Knowing that, you can build a CMS machine to help you get the right stuff to the right people in the right way.

In this white paper I put a definition around the phrase *content management*, relate it to the very young industry with the same name, and link it to some of the Web technologies that you perhaps are now using to deal with sites that have gotten out of control.

Defining Content Management

If you've previously heard about content management, it's most likely because you have connections to a large Web-development project. Today, that's where most of the interest and activity lies. As the Web moved past small, informally designed sites and into large, rapidly changing sites, the need for strong management tools became pressing. Product companies moved in to address this need and called their offerings *content management systems* (CMSs). If your only problem is to create and maintain a large Web site, then you have reason enough to desire the strict structure and formal procedures of a CMS. Such a system helps you get, and stay organized, so that your site can grow and change quickly while maintaining high quality. The Web, however, is simply one of many outlets for information that organizations need to organize. And, as the amount of information sharing between these outlets grows, the desire for an organized approach becomes an absolute need.

I've been giving talks and running seminars on content management for the last couple of years. I often ask my audiences what publications they're responsible for. The very large majority of the

responses once were, "Web site only." An ever-rising number, however, now say, "Multiple Web sites and print publications and anything else that we can create from the same information."

At the highest level, content management is the process behind matching what *you* have to what *they* want. You're an organization with information and functionality of value. They're a set of definable audiences who want that value. This definition and the processes behind it work as well in other outlets as on the Web.

At first blush, content management may seem a way to create large Web sites, but on closer examination, it's in fact an overall process for collecting, managing, and publishing content to any outlet, as the following list describes:

- ⚡⚡ **In collection:** You either create or acquire information from an existing source. Depending on the source, you may or may not need to convert the information to a master format (such as XML). Finally, you aggregate the information into your system by editing it, segmenting it into chunks (or components), and adding appropriate metadata.
- ⚡⚡ **In management:** You create a repository that consists of database records and/or files containing content components and administrative data (data on the system's users, for example).
- ⚡⚡ **In publishing:** You make the content available by extracting components out of the repository and constructing targeted publications such as Web sites, printable documents, and e-mail newsletters. The publications consist of appropriately arranged components, functionality, standard surrounding information, and navigation.

If content management were only a process for creating large Web sites, then using it for organizations with large Web sites would prove well worth the effort. For organizations with multiple, large publications (even if they're multiple Web sites), content management becomes a necessity. A deeper reason exists, however, as to why content management is important. It's not just the common forms of information that I call content that you must collect, manage, and deliver. It's all business. As organizations begin conducting their business electronically, they open the same Pandora's box that they open as they begin delivering information electronically - namely, one that invokes the following questions:

- ⚡⚡ How do I break my business down into electronically deliverable parts?
- ⚡⚡ How do I make sure that I know what parts I have and that these parts are the right ones for my staff, partners, and customers?
- ⚡⚡ How do I ensure that the right part reaches the right person at the right time?

If content management is the process of collecting, managing, and publishing content, then e-business is the process of collecting, managing, and publishing parts of your business. Of course, much of that business entails publishing information. So, with no stretch at all, you can see how content management may underlie some e-business. But organizations conduct business by providing for actions as well as for information. Actions, in an electronic world, take the form of computer functionality. Functionality, in turn, takes the form of objects and blocks of computer programming code, which you distribute in exactly the same way as you do run-of-the-mill content.

The Content Management Industry

At the beginning of the year 2001, the industry of content management is still very new. No standards are universally accepted for what content management systems are or do. Based on my own definition, however, you'd find that the majority of content management systems in use today were created by programmers and Webmasters who were simply trying to keep up with their explosive sites. In the last few years, however, several commercial applications have been released that perform content management in one way or another.

As time goes on, the concepts around content management and the standard offerings of content management systems are sure to coalesce. But for the present, drawing one-to-one comparisons between products is difficult, and clearly stating the advantage of one of these products over a system that you create on your own is often hard. In this white paper, I don't look at particular content management products. If I did, the information that I'd present would be out of date before I could finish writing it. Rather, I build a set of concepts and the framework of the discipline of content management that you can use to decide what you need in your situation, independent of any particular product. Then you can go to the products that are available, with an independent and thorough assessment of what they need to accomplish for you.

Today, you can't find a commercial system that does everything that I present in this white paper. Most major CMS products offer some capabilities in the three areas of collection, management, and publishing, but they concentrate most of their strengths in only one of these areas. In addition, because no center to the discipline of content management exists, any company offering a CMS product is free to define the term *content management* as it sees fit to best appeal to its customers. As time goes on, this situation is certain to change. What isn't likely to change, however, is your responsibility to understand what you're trying to accomplish with a CMS, and how to organize your content enough to benefit from one. That's what I hope that this white paper teaches you.

Note

In addition to providing solid concepts and advice to people working with content management "in the trenches," I also hope that this white paper helps move the content management discipline and industry toward a unified set of concepts and a clear target for the range and depth of functionality that it must provide to truly serve the needs of its clientele.

Do all these qualifiers mean that you should avoid buying a commercial product? That depends. With a little help and customization, the products available today are certainly up to the task of managing a large Web effort. In addition, the CMS products aren't the only ones that are immature and growing. That's the case with the entire digital-information industry. So, if you can afford to wait for the entire "online information" phenomenon to settle down, then you're sure to have an easier time getting a CMS going. But, be willing to wait quite a while. If, like most of the organizations that I know, you've already waited far too long and the weight of your own information is crushing you, then you may as well buy a product and start now. Simply understand that you're signing up for more than a single purchase. You are signing up for participation in the ongoing definition of digital publication technologies.

Systems for Managing Content

People use a very wide variety of tools and methods to process information and get it into publications. From using paper note cards to installing massive database applications, organizations find ways of organizing and managing information. My goal in this white paper isn't to say which of these systems are and aren't true content management. Rather, my goal is to describe the full range of possibilities at your disposal for constructing a system to most effectively handle the range of issues that you may confront.

I begin with the simplest system I've seen that still bears some of the earmarks of a full content management system. I then move up in complexity to describe other systems that embody more and more of the full picture.

The nominal Web CMS

As people started creating Web sites, they did so by typing HTML into plain text editors (such as Microsoft Notepad). As time moved on, the need for better tools grew (fueled by enterprising product companies, less technical users, and the need to automate tedious tasks). At first, these new HTML authoring tools did little more than help you remember the arcane syntax of HTML.

Later, they began to be true WYSIWYG (what you see is what you get) environments for creating Web pages. Today, some of these tools add just enough management to serve as a nominal Web CMS for organizations with small sites and no additional publications.

Tools such as Macromedia Dreamweaver and Microsoft FrontPage intend to serve as the single tool from which you not only create pages, but also share resources between pages, and manage the layout and organization of your site. The main functionality that they offer for these management tasks is as follows:

- ⚡⚡ **Page templates:** These features enable you to create standard page layouts and apply them across pages. Templates enable you to share resources such as images and standard text blocks and to auto-generate some basic navigation based on the pages you add to your site.
- ⚡⚡ **Basic status:** Such functions enable you to track where a page is located and to perform basic workflow functions such as routing pages to the next person that needs to work on them.
- ⚡⚡ **Site outlines and link managers:** These features enable you to see the general relationships between sections or pages and verify that your cross-references are valid.
- ⚡⚡ **Deployment managers:** These functions enable you to upload the site you create locally to the Web server where it's publicly available.

Today's Web-authoring tools (including certainly many more than just Dreamweaver and FrontPage) are making their first halting steps into the realm of content management. I have no doubt that, as time moves on, they will expand their tools to cover more and more of the territory. Still, to really enter the world of content management, these tools must change some of their fundamental assumptions to alleviate the limitations that I describe in the following list:

- ⚡⚡ **The existing tools are small-scale by design.** They assume a small and loosely organized team producing pages one at a time. A CMS assumes a large, very well organized team producing content that will be moved onto pages in bulk.
- ⚡⚡ **They're page oriented.** Managing pages is convenient with a single site at a small size. But, after your site grows to a larger size - and especially if you must create more than just a single site - you need to move beyond managing pages to managing content components.
- ⚡⚡ **These tools assume that you're creating one Web site.** (Or, if you're creating multiple Web sites, they assume ones that are completely independent of each other.) They offer no capability to share the same content between pages of the same site, let alone between pages of two different sites. They're not even close to capable of creating any publication that's not in HTML.

Page templates

Microsoft FrontPage, for example, provides a page-templating function that illustrates a number of the basic ideas behind publication templates. There is a screenshot of a page that I created in a few clicks by using the site-creation wizards in FrontPage 2000 in Figure 1.

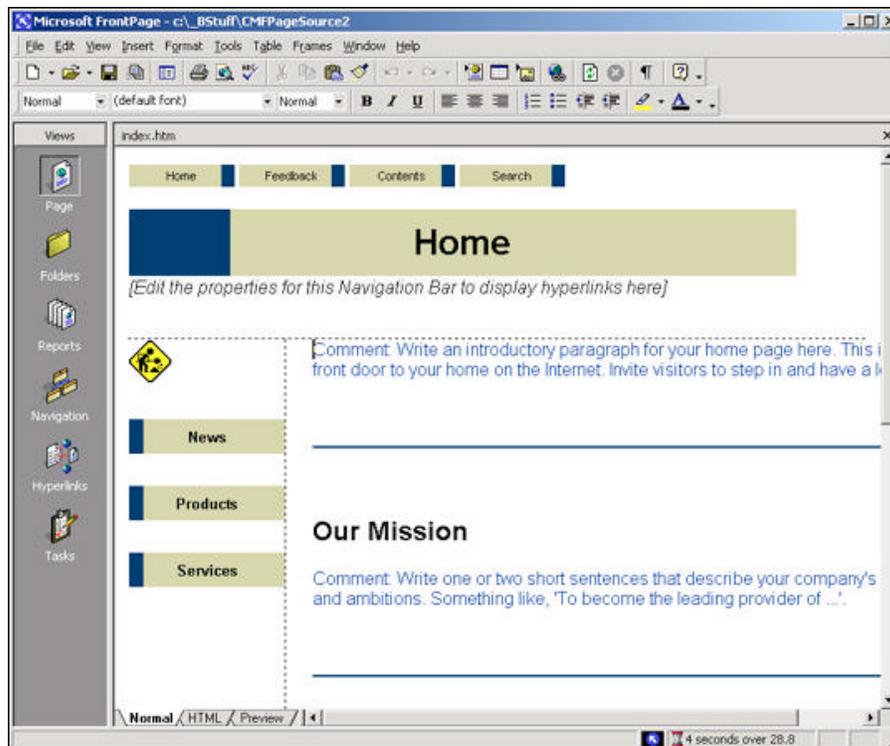


Figure 1: Microsoft FrontPage 2000 page view

The page has everything on it that you may see on a published page except the actual content. In fact, it's a WYSIWYG template for creating pages for this site. The template includes the following features:

- ✂ ✂ A banner and global navigation buttons at the top, with names that the application changes as you change the name of the site or its pages.
- ✂ ✂ A local navigation bar on the left, with button names and order that the application changes as you add or modify pages in this section of the site.
- ✂ ✂ A footer at the bottom with standard text and links that you want to appear on each page.

Although this system is far from a full publication template system, it shows many of the basics. In particular, it shows how the system can specify page layout, page names, and standard content blocks once, and then have the system automatically update and fill in the right stuff in the right place. Most important, if you change the design of the template, the system automatically propagates your change to all affected pages.

Status functions

Macromedia Dreamweaver, for example, provides a function by the name of Design Notes that enables you to specify and review page status and notes (see Figure 2).

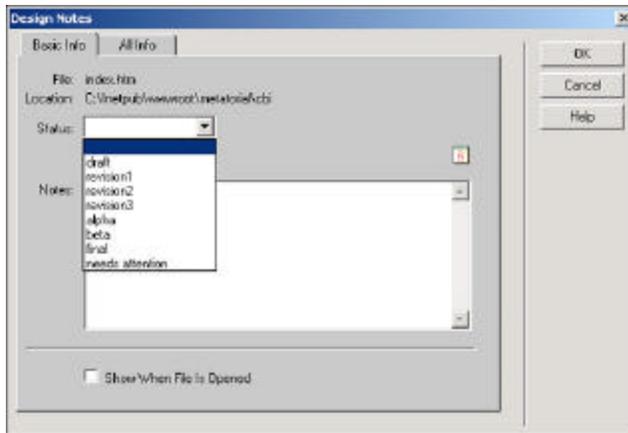


Figure 2: The Design Notes dialog box in Macromedia Dreamweaver 3.0

The box enables you to choose a status for the page you're working on and type whatever notes you want. To help you encourage people to update statuses, the application can display this dialog box each time that the page opens.

This functionality is the mere hint of a workflow and routing system, but it shows the basic idea of tagging pages with management information that you don't publish but that authors and administrators use.

Outlining and organizing functions

Macromedia Dreamweaver provides a site outline and link checker that help you organize the outline and cross-references on your site (see Figure 3).

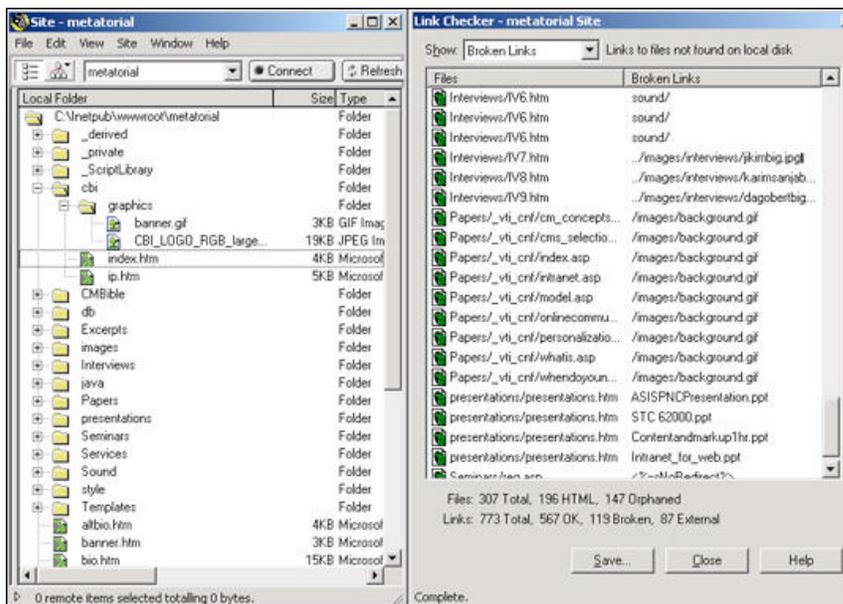


Figure 3: The file view of the site outline and the link checker in Dreamweaver 3.0

A full CMS provides a much more thorough set of tools for managing hierarchies and cross-references (as well as indexes and sequences). The capability to see the directories where your files are stored and to identify broken links is, however, a start.

Deployment tools

Microsoft FrontPage offers a function for deploying your site to a final Web server, where your site becomes "live" on the Web and viewable by the world. There is a screenshot of the Publish Web deployment tool in Figure 4 in FrontPage.

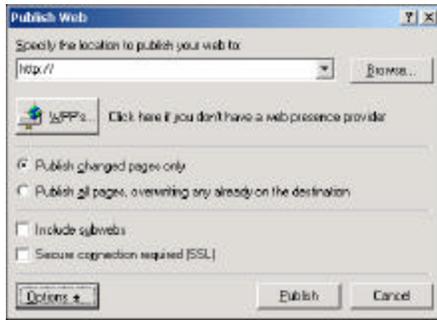


Figure 4: The Publish Web dialog box in FrontPage 2000

This particular function enables you to communicate via FTP (File Transfer Protocol) to a server on the Internet, and deploy all your pages or only those that have changed since the last time you deployed them. This method of file transfer is the most basic form of deployment. I illustrate more advanced forms of deployment in the sections that follow.

The dynamic Web site

A dynamic Web site, sometimes known as a database-driven site, is a system for producing Web pages "on the fly" as users request them.

A data source (a relational database or possibly an XML structure) on the Web server receives a query in response to a user clicking a link. The link activates a template page. The template page contains regular HTML as well as programming scripts, objects, and other programs that interpret the request, connect to the data source, retrieve the appropriate content, and do whatever processing is necessary to form an HTML page. After the template creates the appropriate HTML page, the Web server sends it back to the user's browser (see Figure 5).

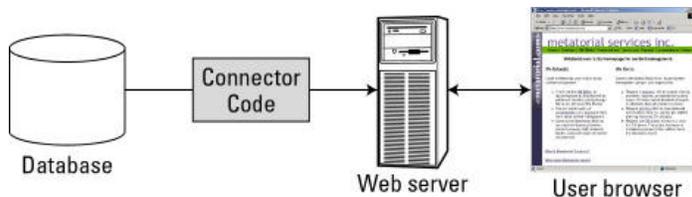


Figure 5: A dynamic Web site creates pages programmatically, sending them back to the inquiring browser.

In a purely dynamic site, no HTML files exist - only the capability to build them whenever someone wants them. Contrast this fluidity with a static site, where all the pages are pre-built and stored on a Web server as HTML files (see Figure 6).

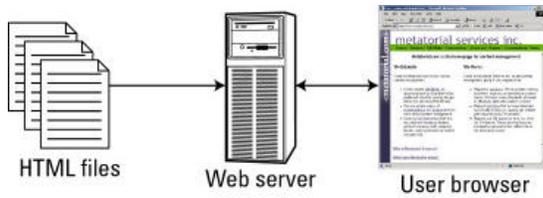


Figure 6: A static Web site delivers pages that are already built.

Given this definition of dynamic sites, you can easily see why people often confuse them with content management systems. For one thing, you can say that content management systems do essentially the same thing. They, too, maintain databases or XML structures, retrieve appropriate content, and return built pages. On the other hand, you have compelling reasons to distinguish between the two. Why? Because you can have a dynamic site that really isn't doing content management at all. In addition, a content management system can just as easily build a static site.

Suppose, for example, that you maintain a large, dynamic Web site that uses advanced scripts to put a user interface on your organization's financial system. The system responds to user requests and dishes out just the right HTML page in response. You'd be hard pressed to call this setup a content management system - it's really just a Web-based application.

You can, on the other hand, have a CMS that produces a static site. Suppose that I set up a very complex CMS that contains millions of components, offers a sophisticated workflow, and produces 100 distinct publications. One of those publications is a Web site. After I hit the right button, out flows a static Web site of one million HTML files. I put those files on a Web server and I'm done. If I want to change the site, I hit the button again, and out flows a new, static Web site that replaces the first one. In this case, I have a robust CMS producing a static Web site (see Figure 7).

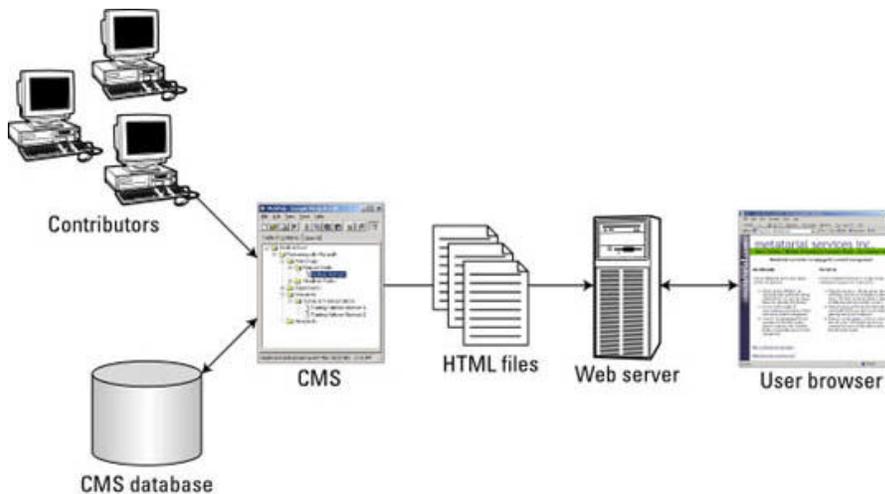


Figure 7: A CMS can, and often does, create static Web sites.

So, dynamic Web sites and content management systems aren't the same, and you don't need to produce a dynamic Web site from a CMS. In fact, if you can get away with only producing static Web sites from your CMS, then you're better off. A static site is faster and much less prone to crashing than a dynamic one. But, you rarely can get away with static sites or, at least, not entirely static ones.

You need a dynamic site if you don't know beforehand what may appear on a page. If you must assess user input or some other factors to figure out what belongs on a Web page, you need a dynamic system. For any realistic degree of personalization, for interaction with data systems (transactions, catalogs, and so on), or for "live" updates (such as changing news stories or stock quotes), you need the processing power on the Web server that a dynamic system gives you.

Although dynamic sites aren't content management systems, they illustrate a number of the qualities of a CMS, as follows:

✍ ✍ **The template pages** on a dynamic site are very similar in approach to the templates in a CMS. Unlike the templates system in some of today's Web-authoring tools, which may be isolated from you and offer a limited set of features, dynamic sites use generalized Web programming technologies such as Java Server Pages (JSP) and Active Server Pages (ASP). These programming languages are virtually unlimited in scope and can create any sort of page layout and logic that you may want. The templating system in many commercial CMS products is no more than an enhanced version of the kinds of template files that dynamic sites use.

✍ ✍ **The types of data sources** on a dynamic site are similar to, and sometimes even the same as, those you use in a CMS. In either a dynamic site or a CMS, you're likely to use the same database or XML products, use the same programming techniques, and write the same sorts of content selection, layout, and navigation-building code. A good CMS, however, provides you with enhancements that make these tasks much easier. CMS data sources also tend to store more management information (metadata) than the data sources for dynamic sites.

What's typically most often missing from a dynamic site system is the capability to create more than just a Web site, and the collection system that a CMS includes. Of course, nothing is stopping the enterprising programmer from extending Web-site code to perform these functions, but then she's not creating a dynamic Web site; she's creating a CMS!

The full Web CMS

Sites don't need to be 100 percent static or 100 percent dynamic. In fact, the vast majority of large Web sites are a little of both. Parts of your site can consist of HTML files and other parts you can dish up dynamically out of a database. In addition, a variety of databases may provide different parts of your site (see Figure 8).

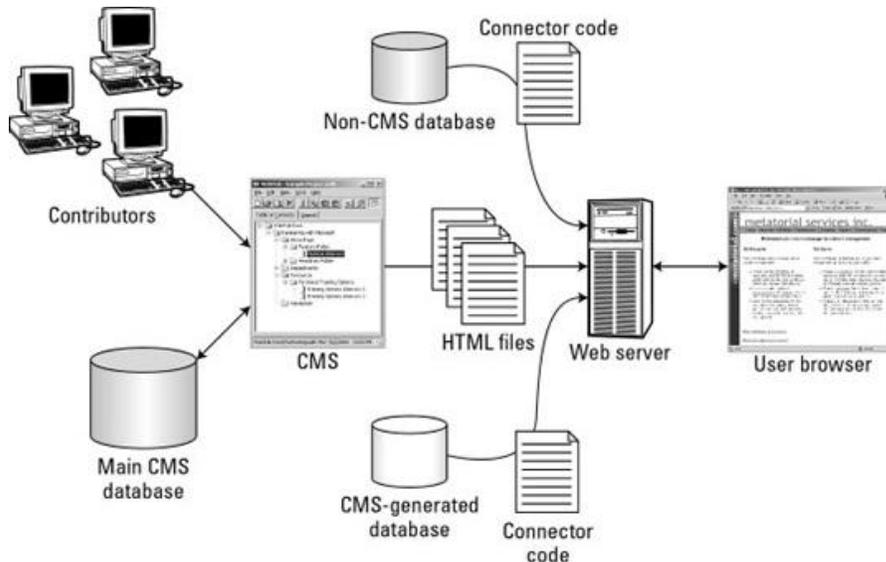


Figure 8: A CMS can stand behind all the static and dynamic parts of your site.

In a full Web CMS, you can have any or all of the following:

- ⚡ **A CMS application:** Sitting safely behind the Web server, this application takes care of collecting content from contributors and managing your content's workflow and administration. The actual architecture varies by product. In some CMS products, you have software on the organization's local area network (LAN) - inside the firewall - for collecting and testing your content, and software outside the organization's LAN, on the live Web server, for dynamically serving up pages.
- ⚡ **A repository:** Also behind the Web server is a relational or XML data source. The repository holds all your content, administrative data, and any of the resources that you need to build the site (such as graphics and style sheets).
- ⚡ **A set of flat HTML files:** The CMS manages and deploys files to the static part of the site.
- ⚡ **A live data source (the CMS-generated database):** This one is located on the Web Server for the dynamic parts of the site. The CMS can deploy data and content from its repository to the CMS-generated database. In this way, even dynamic content can be managed behind the firewall and kept off the server if it is not ready to be seen publicly. In addition, the template pages that access the CMS-generated database can be pages that are created by the CMS.
- ⚡ **Other data sources:** You can connect other sources of data to the Web site that you don't connect to the CMS. A transaction database for conducting sales on the site, for example, you may connect to the Web site but not to the CMS. The other data sources can run completely independently of the CMS (or the template pages) that accesses the sources.
- ⚡ **Templates:** A set of publication templates moves data from any source to the state it needs to be in for the site. In some cases, the data needs to be put into finished static HTML pages. In other cases, it needs to be put into databases that will live on the Web server.

As you can see, you can quickly get a Web site that is pretty complex. It is complex in two ways. First, it has a lot of software and hardware parts that must be coordinated. Second, it has complex information structures that must be stored and correctly delivered. You can manage the complexity of the information with a CMS. If you consider the various chunks of code in the template pages as chunks of functionality (which they are), you can treat them as just another kind of content to collect, manage, and publish onto the right pages.

Tip

In general, your best bet is to drive as much of your site as is feasible toward static pages. Even if content changes once a day, you're probably better off producing it as HTML files and posting the ones that change once a day to your server. Flat pages are tremendously faster and more reliable than dynamic pages. You need compelling reasons to dynamically generate pages - for example, the content is changing minute-by-minute, or you need to personalize the content for each user or user type.

The Enterprise CMS

Most discussions of content management today center around creating a large Web site. Although large Web sites are the primary use of content management systems at the moment, the potential for a CMS to help an organization goes far beyond the Web.

Clearly, a CMS can do a lot more than produce a Web site. It can encompass your entire content creation and organization system. It can provide a content repository where you can review and work on information independently of any page it may land on, and most important, a CMS can produce Web sites and any other publication that you may care to make from the stored content.

If you look at your content collection, management, and publishing needs from the perspective of the whole organization, you quickly see that the potential for content management goes way beyond the Web. In my experience, this insight is dawning only slowly in organizations that are hard-pressed to even get their sites organized, let alone revamp the information processing in their entire enterprise.

Many organizations are accepting (at least for the present) that their print publications are quite separate from their Web publications (not that most would even call a Web site a publication) in the following ways:

- ✍✍ They're willing to spend the extra time and money duplicating effort between a print team and a Web team.
- ✍✍ They're willing to put up with the lack of synchronization between the two sources of information that they produce.
- ✍✍ They're willing to let one publication wait for content until the other goes to press. At first, the Web site was always what waited. More and more, however, the print publication must now wait for Web content.

I think this reticence to think globally is often driven by the entrenchment of print-based teams in their traditional tools and methodologies. For some print team members, the CMS attitude (away from a specific publication and toward creating good content that can serve multiple purposes) is a godsend that enables them to reach new heights. For others, it's no more than needing to learn a lot of new programs and unlearn a lot of old habits.

Not surprisingly, I've seen much more desire to think globally from those who must share content between Web sites or between the Web and other digital devices (wireless phones, for example). These teams already embrace the new authoring tools, content formats, database technologies, and templating models that dynamic Web sites have taught them. Thus, considering going the final steps to a CMS isn't such a big stretch.

Interestingly, the divide between publications on handheld digital devices and the Web is much wider than the divide between print publications and the Web. Handheld appliances can show only a couple of sentences of text at a time and only very, very small pictures. They require a major rethinking of how you deliver anything you're normally used to seeing either in print or on the Web.

In this white paper, I talk most about enterprise content management. Although the Web still figures most prominently, nothing's exclusive about the Web's demand for well-managed content. If your sole concern is the Web, it likely won't stay that way for long.

Summary

Content management systems have a simple purpose - to make it so that you can get content with the least hassle, so that you know what content you have, and so that you can get it out to a variety of places automatically. The details are less simple, as the following list illustrates:

- ✍✍ There is now only the beginning of a content management industry with only immature products currently available.
- ✍✍ Most content management systems in place today cover only the Web.
- ✍✍ Web-authoring tools are great for small sites, but they lack the functionality or scalability to serve as real systems.
- ✍✍ Today's most-sophisticated dynamically generated Web sites go a long way toward content management, but are still much too narrowly focused to do the whole job.

While the technology matures, you're best off spending your time and energy taking what steps you can to move toward a content management approach. This strategy means first figuring out in detail what you want to happen and then seeing how far you can get with today's tools.