

Content Mismanagement Systems

Warren Harrison

In days long ago, when you opened a Web page, it very concretely mapped to a specific file, located at a specific place in your file system. For example, in 1996, if you had opened up my home page at www.cs.pdx.edu/~warren/index.html, your client would have asked my server to deliver the contents of the file at `/u/home/warren/`



`public_html/index.html`. However, the Internet has since moved from small collections of personal Web pages, carefully crafted in native HTML by computing enthusiasts using `vi`, to organizations trying to manage hundreds if not thousands of documents. Not surprisingly, this approach didn't scale.

Content management systems

The software industry—always eager to make a buck, always looking for vacuums in the “better mousetrap” space—answered this question by producing content management systems. With CMSs, you no longer have to painfully type your content into a file with HTML tags. Your content doesn't even have to be in a file—you can just populate some fields in a database, and pow! The CMS produces a Web page on the fly that looks just like ones handcrafted by humans. A Web page generator takes an index into the database (say, a totally meaningless string like `XPLMKJH`), retrieves the content for that index, and uses it to

populate the Web page it's going to create. Literally dozens of such systems exist; a list of open source CMSs can be found at www.la-grange.net/cms.

This is a really slick solution for Web sites that have lots of frequently changing items without a huge amount of unique content. For instance, online catalogs might contain a stock number, a description, and a price for each item. It seems pointless to create a separate HTML file for each item; it makes much more sense to add this entry to a database and then just extract these items and display them on a preformatted Web layout.

URLs of mystery

This approach has led to what I call “URLs of mystery.” In the simplest example, the URL consists of the name of the Web page generator (say a CGI or PHP application), followed by a parameter listing the database index—for instance, `www.servestuff.com/buildpage?index=XPLMKJH`. The main problem with this is that you really can't tell what's on that page by just looking at the URL. So if I bookmark that link, when I return the next day to see what I bought the night before, how do I know what this jumble of characters points to?

As we all know, the answer is the page title—the string in the upper-left-hand corner of your browser that should be populated from the content database to identify the page contents. The title gets saved in your bookmarks along with the URL, theoretically making it

DEPARTMENT EDITORS

Bookshelf: Warren Keuffel,
wkeuffel@computer.org

Design: Rebecca Wirfs-Brock,
rebecca@wirfs-brock.com

Loyal Opposition: Robert Glass,
rglass@indiana.edu

Open Source: Christof Ebert,
christof.ebert@alcatel.com

Quality Time: Nancy Eickelmann,
nancy.eickelmann@motorola.com,
and Jane Hayes, hayes@cs.uky.edu

Requirements: Neil Maiden,
N.A.M.Maiden@city.ac.uk

Tools of the Trade: Diomidis Spinellis,
dds@aub.gr

STAFF

Senior Lead Editor
Dale C. Strok
dstrok@computer.org

Group Managing Editor
Crystal Shif

Senior Editors
Shani Murray, Dennis Taylor, Linda World
Staff Editor Editorial Assistants
Rita Scanlan Brooke Miner and Molly Mraz

Magazine Assistant
Hilda Hosillos, software@computer.org

Art Director
Toni Van Buskirk

Technical Illustrator
Alex Torres

Production Artist
Carmen Flores-Garvey

Executive Director
David Hennage

Publisher
Angela Burgess
aburgess@computer.org

Associate Publisher
Dick Price

Membership/Circulation Marketing Manager
Georgann Carter

Business Development Manager
Sandra Brown

Senior Production Coordinator
Marian Anderson

CONTRIBUTING EDITORS

Robert Glass, Joan Taylor

Editorial: All submissions are subject to editing for clarity, style, and space. Unless otherwise stated, bylined articles and departments, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *IEEE Software* does not necessarily constitute endorsement by the IEEE or the IEEE Computer Society.

To Submit: Access the IEEE Computer Society's Web-based system, Manuscript Central, at <http://cs-ieee.manuscriptcentral.com/index.html>. Be sure to select the right manuscript type when submitting. Articles must be original and not exceed 5,400 words including figures and tables, which count for 200 words each.

easy to figure out which URL points where. However, page titles aren't always as helpful as you might think. One of my frequent online shopping sites lists just the company's name in the title for every item you display, so at the end of the day you end up with a list of bookmarks that all have the same title and an undecipherable URL.

Email me a URL, go ahead, I dare you!

One of my favorite pastimes is emailing friends URLs of pages I happen to come across while browsing the Web. In Mozilla (my browser of choice), it's easy enough: simply right-click on the page and select "send link." My email client generates a message containing the URL to the page in question and gives me an opportunity to add some notes before sending it off to the lucky recipient.

However, as CMSs have become more ubiquitous, this has become harder and harder to do. Recently, I tried to send a potential author a link to our Author Guide. Shockingly, the URL on that page—www.computer.org/portal/site/software/menuitem.538c87f5131e26244955a4108bcd45f3/index.jsp?&pName=software_level1&path=software/content&file=write_for.xml&xsl=article.xsl&—is 170 characters long. Most mail clients end up wrapping the URL across several lines on the receiving end, requiring the recipient to do major surgery on the string just to load the page. Oddly enough, the file this monstrosity points to is actually located at www.computer.org/portal/pages/software/content/write_for.html—still 61 characters long, but that's a far cry from 170!

Why should we care?

Most organizations that use CMSs find that viral marketing is an important part of their business strategy. Viral marketing is the high-tech version of "word of mouth." I find something about your Web page I like, I send it to three friends, they each send it to three of their friends, and before you know it, that one page view has resulted in dozens, if not hundreds, of page views.

What happens to this concept if I end up having to email a 170-character URL to my three friends? Maybe one of them chooses to do the surgery and stitch the URL back together, but of course their three friends get the URL wrapped across three lines as well. And maybe none of them feel like doing URL surgery. Your message is lost.

Many Web site publishers obviously recognize this. It's pretty common to see a link that says "send this URL to a friend." Well, fine then. Maybe I'll go to the trouble of filling in the form that pops up when I press that button—maybe (with Amazon.com I actually have to sign into their secure server with my email address and password, and navigate several different pages just to send a page to a friend). And maybe my friend's spam filter will accept that message from honestjoes.com that contains the URL I just caused to be sent—maybe. And maybe my friend will open that message, even though it's from someone they've never heard of before—maybe. But I wouldn't want to bet my business on any of these.

How do your users use your application?

The moral to this story is that designers often get wrapped up in technical decisions and believe they won't affect system users, but they're wrong. There are plenty of examples of these user-unfriendly CMSs out there. For example, one e-commerce site I visited recently when looking for a pair of binoculars fills my browser's address field with the following URL: www.cheaperthandirt.com/ctd/mixeddept.asp?dept%5Fid=18&dept%5Fname=Optics&msscoid=3NBQD1X9XJ4J8K9SGE5MHECDBH66S2. When shopping for a Christmas gift for my wife, Amazon.com returned this unintelligible 117-character URL: www.amazon.com/gp/product/0452283442/qid=1133738553/sr=8-2/ref=pd_bbs_2/104-6933329-0354319?n=507846&s=books&v=glance. A link on *Scientific American's* Web site promising to tell me how GPS devices work looks like this: www.sciam.com/askexpert_question.cfm?articleID=000349D4-D6FC-

1CFC-93F6809EC5880000&catID=3&chanID=sa005. If I want to check out what eBay is auctioning in the way of Macintosh computers, the 133-character URL to get me there is computers.listings.ebay.com/Apple-Macintosh-Computers_W0QQfromZR4QQsacatZ4599QQsocmdZListingItemListQQss-PageNameZdcpComputersTextFeat.

Do these URLs *have* to be this long? Given today's content management systems, unfortunately the answer appears to be "yes." But with some judicious design decisions, we can minimize gratuitous URL padding.

More friendly URLs

To illustrate, let's dissect the Author Guide URL I mentioned earlier: www.computer.org/portal/site/software/menuitem.538c87f5131e26244955a4108bcd45f3/index.jsp?&pName=software_level1&path=software/content&file=write_for.xml&xsl=article.xsl&. How many choices did this site's designers have when laying out the content architecture? The first thing that jumps out at me is the choice of a directory structure that's four levels deep. Is it necessary to partition the content into separate directories? I don't know. It's probably important to separate *IEEE Software* articles from those in *IEEE Transactions on Software Engineering*, but does "portal" really need subdirectories?

The second thing is the obvious selection of a 128-bit hash of (I assume) the file name so you end up with a unique string for each file and avoid collisions. But was it necessary to append "menuitem" to that mess? It's also good to remember here that if someone is inclined to stitch your URL back together, the place where humans have the most trouble is in reassembling random collections of characters. Guess where my email client breaks the URL when it receives it.

Once you get past the hash, we see that the Web page generator (index.jsp) is being sent four parameters: name, path, file, and xsl. How many of these are necessary? Can't the server be configured to default to an "index" page? The URL has already placed us in the

"software" directory; do we need a "path" parameter? How about "xsl"? A quick look at the Web site shows that almost all content seems to have an "xsl" of "article.xsl." Is this really necessary?

Contrast this URL (and the other URLs I listed earlier) with the URL to a *PC Magazine* review of a laptop I am evaluating: www.pcmag.com/article2/0,1895,1889060,00.asp. This is simplicity itself. Only 44 characters, and of that 13 of them are for the site's domain name.

Making implementation decisions

I teach my students that when designing systems, there are *requirements*, *constraints*, and *goals*. Requirements are those things that can be articulated and that a system must have. Constraints limit the designers' freedom in meeting the requirements. Because both requirements and constraints can be articulated, you can tell if they're there or not. Saying that a CMS must load a Web page from a MySQL database is a requirement (with a constraint thrown in)—I can come up with a test case to make sure it works and check to make sure you're using MySQL.

On the other hand, goals are those amorphous properties that we just can't (or aren't willing to) spec out. Things like making a system fast, reliable, or ... easy to use. Goals are like pornography: I know it when I see it. But goals also provide a basis for decision-making. If the system can be developed in one of two ways and still meet the requirements, goals tell us which way is right. Goals should help us decide if we should stick our content four levels deep, repeat information in URL parameters we already have, and append invariant nine-character strings to 32-bit hashes to construct a 41-character identifier. A system may meet the requirements, but what about the goals?

Producing short URLs

If you've got an unwieldy URL you want to send to people, what can you do about it? Well, there are several excellent resources out there to solve the problem

EDITOR IN CHIEF

Warren Harrison

10662 Los Vaqueros Circle
Los Alamitos, CA 90720-1314
warren.harrison@computer.org

EDITOR IN CHIEF EMERITUS:
Steve McConnell, Construx Software
stemcco@construx.com

ASSOCIATE EDITORS IN CHIEF

Education and Training: Don Bagert, Rose-Hulman Inst. of Technology; don.bagert@rose-hulman.edu

Design: Philippe Kruchten, University of British Columbia; kruchten@iee.org

Requirements: Roel Wieringa, University of Twente; roelw@cs.utwente.nl

Management: Don Reifer, Reifer Consultants; dreifer@earthlink.net

Quality: Stan Rifkin, Master Systems; sr@master-systems.com

Experience Reports: Wolfgang Strigel, QA Labs; strigel@qalabs.com

EDITORIAL BOARD

Christof Ebert, Alcatel
Nancy Eickelmann, Motorola Labs
Jane Hayes, University of Kentucky
Warren Keuffel, independent consultant
Neil Maiden, City University, London
Diomidis Spinellis, Athens Univ. of Economics and Business
Richard H. Thayer, Calif. State Univ. Sacramento
Rebecca Wirfs-Brock, Wirfs-Brock Associates

ADVISORY BOARD

Stephen Mellor, Mentor Graphics (chair)
Maarten Boasson, Quaerendo Invenietis
J. David Blaine, ViaSat
Robert Cochran, Catalyst Software
Annie Kuntzmann-Combelles, Q-Labs
David Dorenbos, Motorola Labs
Kaoru Hayashi, SRA
Simon Helsen, SAP
Juliana Herbert, ESICenter UNISINOS
Dehua Ju, ASTI Shanghai
Gargi Keeni, Tata Consultancy Services
Karen Mackey, Cisco Systems
Tomoo Matsubara, Matsubara Consulting
Dorothy McKinney, Lockheed Martin Space Systems
Bret Michael, Naval Postgraduate School
Susan Mickel, Lockheed Martin
Ann Miller, University of Missouri, Rolla
Deependra Moitra, Infosys Technologies, India
Melissa Murphy, Sandia National Laboratories
Suzanne Robertson, Atlantic Systems Guild
Grant Rule, Software Measurement Services
Girish Seshagiri, Advanced Information Services
Martyn Thomas, Praxis
Rob Thomsett, The Thomsett Company
Laurence Tratt, King's College London
Jeffrey Voas, SAIC
John Vu, The Boeing Company
Simon Wright, SymTech

CS PUBLICATIONS BOARD

Jon G. Rokne (chair), Michael R. Blaha, Mark Christensen, Frank E. Ferrante, Roger U. Fujii, Phillip Laplante, Sorel Reisman, Jon Rokne, Bill N. Schilit, Linda Shafer, Steven L. Tanimoto, Wenping Wang

MAGAZINE OPERATIONS COMMITTEE

Bill N. Schilit (chair), Jean Bacon, Pradip Bose, Arnold (Jay) Bragg, Doris L. Carver, Kwang-ting (Tim) Cheng, Norman Chonacky, George Cybenko, John C. Dill, Robert E. Filman, David Alan Grier, Warren Harrison, James Hendler, Sethuraman (Panch) Panchanathan, Roy Want


Welcome!

We'd like to welcome Rebecca Wirfs-Brock who, with this issue, begins editing the Design column that Martin Fowler managed for the past five years. Rebecca is best known for her role in inventing Responsibility-Driven Design. She is lead author of the classic *Designing Object-Oriented Software* (Prentice Hall, 1990) and *Object Design: Roles, Responsibilities and Collaborations* (Addison-Wesley, 2003). Through her writing, teaching, and speaking, she has popularized the use of informal techniques and thinking tools for designers, analysts, and programmers. Among her widely used inventions are object role stereotypes and the conversational form of use cases. Prior to founding Wirfs-Brock Associates in 1997, she spent 15 years at Tektronix, where her accomplishments included managing the first commercial Smalltalk system and managing the Reusable Components Group at Tektronix's TekLabs. After she left Tektronix she served as director of consulting and training for Instantiations, Digitalk, and ParcPlace-Digitalk. She's on the Agile Alliance's board of directors. You might enjoy visiting her blog at www.wirfs-brock.com/rebeccasblog.html.



for users. For example, TinyURL (tinyurl.com) will create a short URL from a long one. The 133-character eBay URL I mentioned earlier can be compressed to simply tinyurl.com/dudfe. The 170-character *IEEE Software* Author Guide URL is tinyurl.com/9kymy. You can even add tinyurl to your toolbar in some browsers so that you can compress URLs without going to the tinyurl site. Other URL converters include www.digbig.com, shorl.com, and www.snipurl.com. These are convenient solutions to what is becoming a common problem with modern content management systems.

Feedback welcome

What do you think? Do URLs of mystery frustrate you as much as they do me? Are there other solutions to this problem I am missing? Please write me at warren.harrison@computer.org. 

IEEE International Symposium on

Secure Software Engineering

March 13–15, 2006

McLean, VA, USA

<http://www.jmu.edu/iiia/issse>



This new Symposium series covers the latest in research and experience reflecting the community's response to the need for more secure software and addresses the interests of the commercial, governmental, educational, and research sectors. Mark your calendar and plan to attend.

Today, security problems involving computers and software are frequent and widespread with the number and consequences of attacks increasing rapidly. Successful software security is fundamentally a software engineering problem, encompassing producing and evaluating secure software. The Symposium's scope includes processes, techniques, technology, and people, and will bring you the latest interesting and useful developments.

Includes tutorials and precedes the co-located US DHS/DoD Software Assurance Forum.



TFIA
Task Force
on Information Assurance

