

Philip Greenspun's

A comprehensive look at the current state of
Web site architecture

A complete guide to Web site design
using databases

Enjoy the benefits of personalization,
collaboration, and up-to-date site tracking

From the architect of RDBMS backed Web
services for Hearst Corporation



Database Backed Web Sites:

THE THINKING
PERSON'S GUIDE
TO WEB PUBLISHING

"Caught up in the frenzy of just being there, we have all been enticed by the possibilities the Net offers in a one-to-one world of idea and transactional commerce. Many have planted expensive seeds on the Net with the hope that they will germinate into something of value. Greenspun has thoroughly captured the logic of what can win from the viewpoint of both the creator, who may be, and the viewer, who is, king. Information and ideas can serve lofty purposes but without organization, tools to easily mine and find, and an audience to share and view, there is no purpose."

—Mark Miller

Executive Vice President & General Manager, Hearst Magazines

"Philip Greenspun **interfaced** our 60 GB Oracle medical record database to the Web in 1994. It took **him three** weeks and all it cost me was a 4 GB hard disk and some SIMMs for his Macintosh. Since then, **Greenspun has** become too rich and famous to write Perl anymore. I'm going to **use this book** to train some high school kids to replace him."

—Isaac S. Kohane, M.D., Ph.D.

Director, Children's Hospital Informatics Program

"Smart companies are using the Web to support collaboration. Smarter companies will be using the Web to support computer-to-computer transactions. Philip Greenspun has written the first book that explains how to do both elegantly."

—Mort Myerson, Chairman of Perot Systems Corporation

"IBM has been working for 20 years to build high performance relational database management systems. This book gives you field-tested ideas and techniques to unlock the data and deliver it to users across the Internet."

—John Patrick

Vice President of Internet Technology, IBM

"Greenspun's prize-winning and colorful Web publishing experience is as old as the Web and a must for all aspiring Web publishers."

—Michael L. Dertouzos

Director MIT Laboratory for Computer Science
Author of *What Will Be*

"There are a lot of books that will tell you how to design a web site that looks like something else—a computer game, or a print magazine—which would be great if web pages *were* computer games or print magazines. They aren't. Read this book to find out what they are"

—Robert S. Thau, Ph.D., author of the Apache httpd server

"This is the first book on RDBMS-backed Web sites that an intelligent programmer can enjoy."

—Doug McKee, Manager, Web Server Development, America OnLine

**DATABASE BACKED
WEB SITES:
THE THINKING PERSON'S
GUIDE TO WEB PUBLISHING**

DATABASE BACKED
WEB SITES:
THE THINKING PERSON'S
GUIDE TO WEB PUBLISHING

Philip Greenspun



Ziff-Davis Press

An imprint of Macmillan Computer Publishing USA
Emeryville, California

Publisher	Stacy Hiquet
Associate Publisher	Steven Sayre
Acquisitions Editor	Simon Hayes
Development Editor	Paula Hardin
Copy Editor	Kim Haglund
Technical Reviewer	Jin Choi
Proofreader	Jeff Barash
Cover Illustration and Design	Megan Gandt
Book Design	Bruce Lundquist
Page Layout	Janet Piercy
Indexer	Christine Spina

Ziff-Davis Press, ZD Press, and the Ziff-Davis Press logo are trademarks or registered trademarks of, and are licensed to Macmillan Computer Publishing USA by Ziff-Davis Publishing Company, New York, New York.

Ziff-Davis Press imprint books are produced on a Macintosh computer system with the following applications: FrameMaker®, Microsoft® Word, QuarkXPress®, Adobe Illustrator®, Adobe Photoshop®, Adobe Streamline™, MacLink® Plus, Aldus® FreeHand™, Col'age Plus™.

Ziff-Davis Press, an imprint of
Macmillan Computer Publishing USA
5903 Christie Avenue
Emeryville, CA 94608
Copyright © 1997 by Philip Greenspun. All rights reserved.

All other product names and services identified throughout this book are trademarks or registered trademarks of their respective companies. They are used throughout this book in editorial fashion only and for the benefit of such companies. No such uses, or the use of any trade name, is intended to convey endorsement or other affiliation with the book.

No part of this publication may be reproduced in any form, or stored in a database or retrieval system, or transmitted or distributed in any form by any means, electronic, mechanical photocopying, recording, or otherwise, without the prior written permission of Macmillan Computer Publishing USA, except as permitted by the Copyright Act of 1976, and except that program listings may be entered, stored, and executed in a computer system.

THE INFORMATION AND MATERIAL CONTAINED IN THIS BOOK ARE PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY WARRANTY CONCERNING THE ACCURACY, ADEQUACY, OR COMPLETENESS OF SUCH INFORMATION OR MATERIAL OR THE RESULTS TO BE OBTAINED FROM USING SUCH INFORMATION OR MATERIAL. NEITHER MACMILLAN COMPUTER PUBLISHING USA NOR THE AUTHOR SHALL BE RESPONSIBLE FOR ANY CLAIMS ATTRIBUTABLE TO ERRORS, OMISSIONS, OR OTHER INACCURACIES IN THE INFORMATION OR MATERIAL CONTAINED IN THIS BOOK, AND IN NO EVENT SHALL MACMILLAN COMPUTER PUBLISHING USA OR THE AUTHOR BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF SUCH INFORMATION OR MATERIAL.

ISBN 1-56276-530-2

Manufactured in the United States of America
10 9 8 7 6 5 4 3 2 1

CONTENTS AT A GLANCE

1	Envisioning a Site That Won't Be Featured in suck.com	1
2	So You Want to Join the World's Grubbiest Club: Internet Entrepreneurs	12
3	Learn to Program HTML in 21 Minutes	32
4	Adding Images to Your Site	44
5	Publicizing Your Site (Without Irritating Everyone on the Net)	66
6	So You Want to Run Your Own Server	80
7	Learning from Server Logs	110
8	Java and Shockwave—The <BLINK> Tag Writ Large	132
9	Sites That Are Really Programs	148
10	When Is a Site Really a Database?	168
11	Relational Databases	180
12	Interfacing a Relational Database to the Web	214
13	Case Studies	252
14	Sites That Don't Work (And How to Fix Them)	308
15	A Future So Bright You'll Need to Wear Sunglasses	332
	Index	354

TABLE OF CONTENTS

Chapter 1	Envisioning a Site That Won't Be Featured in suck.com	1
	You Can't Say "Web Publishing" without the Word "Publishing"	1
	Example 1: Personal Home Page	2
	Example 2: Camera Manufacturer	3
	Example 3: Software Company	3
	How Do You Know When You Are Done?	4
	Become Illiterate (That Is, Present Multiple Views)	4
	Think of the Web as Primary	7
	Summary	11
Chapter 2	So You Want to Join the World's Grubbiest Club: Internet Entrepreneurs	12
	The Steam Engine and the Railroad	14
	Four Ways to Make Money from Your Site	15
	Travel Example	22
	Real Estate Example	23
	Medical Example	23
	We Lose Money on Every Hit But Make It Up on Volume	24
	A Final Plea	27
Chapter 3	Learn to Program HTML in 21 Minutes	32
	"One of Our Local Webmasters"	33
	You May Already Have Won \$1 Million	34
	Document Structure	35
	Now That You Know How to Write HTML, Don't	38
	It's Hard to Mess Up a Simple Page	39
	Why Graphic Designers Just Don't Get It	41
	Summary	42

Chapter 4 Adding Images to Your Site	44
Are Net Users Really That Shallow?	45
Images on the Web Can Look Better Than on a Magazine Page	46
Start By Thinking about Building an Image Library	47
Using Kodak PhotoCD to Manage Your Image Library	47
Delivering Your Library to the Web	49
If It Worked You Wouldn't Want It	49
In Any Case, It Doesn't Work	49
Maybe You Want Flashpix	50
Choosing between GIF and JPEG	53
Creating GIF and JPEG from PhotoCD Image Pacs	53
Using Photoshop	53
Using Unix Tools	56
Organizing JPEGs on Your Web Server	61
Adding Images to Your Web Pages	62
Summary	63
 Chapter 5 Publicizing Your Site (Without Irritating Everyone on the Net)	 66
Search Engines	67
How Search Engines Look to the User	67
How Search Engines Work	68
Component 1: The Crawler (or "How to Get Listed by a Search Engine")	68
Component 2: The Full-Text Indexer	70
Component 3: User Query Processor	71
How to Stand Tall in the Search Engines	71
How Many Users Are You Getting from Search Engines?	72
Improving Your Pages' Chances Honestly (and Dishonestly)	74

And Now the Dishonest Part	75
Hiding Your Content from Search Engines (Intentionally)	75
Hiding Your Content from Search Engines (By Mistake)	76
Web Directories	77
Final Tip	78
Summary	78
Chapter 6 So You Want to Run Your Own Server	80
My Personal Choice	82
Choosing a Computer	84
Unix	85
Which Brand of Unix Box?	88
Windows NT	89
Unix versus NT	91
Final Hardware Selection Note	93
Server Software	93
API	93
RDBMS Connectivity	95
Support and Source Code Availability	95
Availability of Shrink-wrapped Plug-ins	96
Speed	97
AOLserver (a.k.a. GNNserver, NaviServer)	97
Apache	99
Netscape Enterprise/FastTrack	99
Oracle WebServer 2.0	103
Connectivity	104
ISDN	104
Your First T1	106
Cable Modems and ADSL	107
The Big Picture	108

Chapter 7 Learning from Server Logs 110

Case Studies	112
"I Want to Know How Many Users Requested Non-existent Files and Where They Got the Bad File Names."	112
"I Want to Know How Many People Are Looking at Chapter 3 of http://webtravel.org/samantha/ ."	113
"I Want to Know How Long the Average Reader of Chapter 3 Spends before Moving on to Chapter 4."	115
"I Sold a Banner Ad to Sally's Sad Saab Shop. I Want to Know How Many People Clicked on the Banner and Went Over to Her Site."	116
"I Want to Know the Age, Sex, and Zip Code of Every Person Who Visited My Site So That I Can Prepare a Brochure for Advertisers."	121
Case Studies Conclusions	125
Let's Back Up for a Minute	125
Enter the Log Analyzer	126
wwwstat	127
WebReporter	128
Relational Database-backed Tools	128
Summary	130

Chapter 8 Java and Shockwave— The <BLINK> Tag Writ Large 132

What Java Is	133
What It Is (Also)	136
What It Is (New Techno Stuff)	138
Too Bad It Doesn't Work	139
The 1970s	139
The 1980s	139
The 1990s	140
The Future So Bright You'll Need to Wear Sunglasses	140
You Need These Things	140

Encouragement	141
Using Java Applets on the Web	141
Animation Can Be Useful	142
When Java Makes Sense	142
Richer User Interface	143
Real-time Response	143
Real-time Updates	143
An Inspiring Example	144
Finally . . . Don't Get Too Excited by the Rich User Interface	146
Chapter 9 Sites That Are Really Programs	148
CGI Scripting	150
A Very Simple Perl CGI Script	151
My Unix Box Does Not Like to Fork 500,000 Times a Day	151
My RDBMS Does Not Like to Be Opened and Closed 500,000 Times a Day	152
Server APIs	152
AOLserver Example: Redirect	153
AOLserver Example: Bill Gates Personal Wealth Clock	155
The Right Way—Extending HTML	161
Server-parsed HTML	162
HTML as a Programming Language	163
Summary	166

Chapter 10 When Is a Site Really a Database?	168
The Hard Part	171
Step 1: The Data Model	171
Step 2: Legal Transactions	171
Step 3: Mapping Transactions onto Web Forms	171
The Easy Part	172
Prototyping the Site, My Theory	172
Prototyping the Site, the Reality	173
Do I Need a College Education to Understand Your System?	173
Why Don't Customers Wise Up?	174
Is There a Better Way?	176
Just Say No to Middleware	177
Summary	178
Chapter 11 Relational Databases	180
What's Wrong with a File System (and Also What's Right)	181
What Do You Need for Transaction Processing?	183
Finding Your Data (and Fast)	184
Enter the Relational Database	185
How Does This RDBMS Thing Work?	187
SQL the Hard Way	193
Brave New World	197
Braver New World	201
Choosing an RDBMS Vendor	202
Paying an RDBMS Vendor	203
Performance	204
Don't Forget to Back Up	206
Reliability	208
If You Hire an ISP, Buy the Right Service	210
What I Did for Myself	210
Summary	213

Chapter 12 Interfacing a Relational Database to the Web 214

How Does an RDBMS Talk to the Rest of the World?	215
How to Make Really Fast RDBMS-Backed Web Sites	219
CORBA: MiddleWare Meets VaporWare	222
Complexity (or "It Probably Won't Work")	223
What If It Did Work?	223
Aren't Objects the Way to Go?	224
Security	225
What Does This Stuff Look Like?	228
Server-side Web/RDBMS Products	233
Oracle WebServer 2.0	233
Informix Web DataBlade	236
Every Page a Transaction	239
Web Blade Workaround 1: EXEC Perl	239
Web Blade Workaround 2:	
Convert to AOL server	239
Web Blade Workaround 3:	
Convert to Meta-HTML	240
Web Blade Bottom Line	240
Netscape LiveWire	241
Meta-HTML	241
WebObjects, Microsoft IIS, and Others	242
AOLserver	243
Choosing	244
Bring in Da Noise, Bring in Da Junk	245
Canned Web Server Admin Pages	248
Spreadsheet-like Access	248
Forms Builders	248
Connecting	249
ODBC	250
Summary	251

Chapter 13 Case Studies	252
Case 1: The Mailing List	254
Step 1: The Data Model	254
Step 2: Legal Transactions	254
Step 3: Mapping Transactions onto Web Forms	255
Step 4: Writing Code to Process the Forms	256
Case 2: The Mailing List	261
Case 3: The Birthday Reminder System	264
Step 1: The Data Model	265
Step 2: Legal Transactions	267
Step 3: Mapping Transactions onto Web Forms	269
Step 4: Writing Code to Process the Forms	269
Step 5: Step 5?	275
Case 4: The Bulletin Board	279
Can 30 Million Usenet Users Really Be Wrong?	281
Step 1: The Data Model	282
Step 2: Legal Transactions	283
Step 3: Mapping Transactions onto Web Forms	284
Step 4: Writing Code to Process the Forms	288
Case 5: The Bulletin Board (Again)	288
Step 1: The Data Model	288
Step 2: Legal Transactions	291
Step 3: Mapping Transactions onto Web Forms	291
Step 4: Writing Code to Process the Forms	292
Case 6: The Bulletin Board (Full-text Indexed)	294
Case 7: Classified Ads	295
Classifieds Idea: Categorize	297
Classifieds Idea: Automatic Management	298
Classifieds Idea: E-mail Alerts	298
Classifieds Idea: Show Off New Ads	298
Classifieds Idea: Auction	298
Steps 1–4	298

The Cover Page	300
The Auction System	306
Summary	307
Chapter 14 Sites That Don't Work (And How to Fix Them)	308
User Feedback	309
Category 1: How to Scan?	316
Category 2: General Photography Questions	316
Category 3: How to Make My Life As Cool As Yours?	319
But What about Comments?	
(Loquacious, Version 1.0)	319
Loquacious, Version 2.0	321
Loquacious, Version 3.0	325
Related Links	325
No Dead Links	327
Instant Updates	327
Spam Resistance	327
Distributed Management	328
How Does It Work?	328
General Lessons	329
Chapter 15 A Future So Bright You'll Need to Wear Sunglasses	332
Should Software Really Be Sold Like Tables and Chairs?	335
The Nub	335
Johnny the User	336
Amanda the User Interface Programmer	336
Joey the Image Editor Programmer	336
Adobe the Software Publisher	337
Choices Summary	337
A Better Way	337
Johnny the User	337

Amanda the User Interface Programmer	337
Joey the Image Editor Programmer	338
Adobe the Software Publisher	338
New World Order	338
A Less Radical Approach	339
Privacy	339
We Have a Network; We Can Do Better	339
Your User's Browser: a GE Range	340
Product Engineering: Theory versus Reality	340
What Kinds of Things Can Happen in a Networked House?	341
What Does This Mean to Me As a Web Publisher?	341
Personalization	342
Reader Ratings: A Big Mistake?	344
Client-side Personalization	346
What Does This Mean to Me As a Web Publisher?	347
Collaboratively Exchanged Data Models	348
Collaboratively Evolved Data Models	351
Grand Conclusion	352
Index	354

ACKNOWLEDGMENTS

Common courtesy demands that I thank all of the people who've generously given their time over the years to educate me about the Web and RDBMS. However, as deadly boring as the average computer book may be, the acknowledgments are inevitably far more boring. I have only ever seen one acknowledgments page with a scintilla of literary value, that for the Scheme Shell Reference Manual. I reprint it here in hopes of starting a refreshing trend toward all computer book authors reprinting this acknowledgment rather than thanking their families, dogs, editors, and so on.

Who should I thank? My so-called "colleagues," who laugh at me behind my back, all the while becoming famous on my work? My worthless graduate students, whose computer skills appear to be limited to downloading bitmaps off of netnews? My parents, who are still waiting for me to quit "fooling around with computers," go to med school, and become a radiologist? My department chairman, a manager who gives one new insight into and sympathy for disgruntled postal workers?

My God, no one could blame me—no one!—if I went off the edge and just lost it completely one day. I couldn't get through the day as it is without the Prozac and Jack Daniels I keep on the shelf, behind my Tops-20 JSYS manuals. I start getting the shakes real bad around 10 A.M., right before my advisor meetings. A 10 oz. Jack 'n Zac helps me get through the meetings without one of my students winding up with his severed head in a bowling-ball bag. They look at me funny; they think I twitch a lot. I'm not twitching. I'm controlling my impulse to snag my 9 mm Sig-Sauer out from my day-pack and make a few strong points about the quality of undergraduate education in Amerika.

If I thought anyone cared, if I thought anyone would even be reading this, I'd probably make an effort to keep up appearances until the last possible **moment**. But no one does, and no one will. So I can pretty much **say exactly what I think**.

Oh yes, the acknowledgments. I think not. I did it. I did it all, by myself.

— Olin Shivers

Cambridge, September 4, 1994

Well, I guess Olin thought it was easier than explaining why he had written a program to let people drive a 23-year-old operating system (UNIX) in a 35-year-old computer language (Lisp)

Envisioning a Site That Won't Be Featured in suck.com

**YOU CAN'T SAY "WEB PUBLISHING"
WITHOUT THE WORD "PUBLISHING"**

**BECOME ILLITERATE
(THAT IS, PRESENT MULTIPLE VIEWS)**

THINK OF THE WEB AS PRIMARY

Chapter

1

This is a book about how to be a nerd. A book about how to make a Web server on a cheap computer serve a million hits every day. About how to write clever programs that collect data from Web users and stuff them into a relational database. About how to add thousands of images to your Web site in a few minutes. About how to get serious about user interface design for Web sites.

Before ascending into Nerd Heaven, though, I thought it was worth two chapters to step back and ask, "Why did the world buy 10 million expensive computers and connect them together?" Was it really so that a kid in Botswana could look at a flashing GIF 89a logo from a Honda repair shop in Sunnyvale, California? And, if we don't think that is true, how should we design sites?

An alternative formulation of this question is "How can I design a site that won't be featured at <http://suck.com>?"

YOU CAN'T SAY "WEB PUBLISHING" WITHOUT THE WORD "PUBLISHING"

When you put up a Web site, you are publishing. Virtually all of the important decisions that you must make are publishing decisions. Eventually you will have to select technology to support those decisions, but that is a detail.

Start by putting yourself in your users' shoes. Why are they coming to your site? If you look at most Web sites, you'd presume that the answer is "User is extremely bored and wishes to stare at a blank screen for several minutes while a flashing icon loads, then stare at the flashing icon for a few more minutes."

Academic computer scientists refer to this process of fitting software systems to people as "user modeling."

Slightly more content-rich sites are based on the user model of "User wants to look at product brochures" or "User wants to look at fancy graphics." After pulling the server logs for the sites that reflect these user models, though, it is tough to have much faith in them.

Think about it for a minute: If a user wanted a flashing computer screen and confusing user interface, he could stuff a CD-ROM into the drive. He could get an even more enticing show without the crummy user interface by picking up his television remote control and flipping channels. If a user wanted product brochures, he could get them by calling manufacturers or visiting shops. If a user wanted fancy graphics, he could flip through dozens of pages' worth in a print magazine in the amount of time it would take to load a single corporate Web page.

Users come to Web sites because they have questions. They are not bored losers. You are not doing them a favor by putting a product brochure online or showing them some huge logo GIF. They are doing you a favor by visiting your Web site. They are paying to visit your Web site, if not exactly with money then at least with their time. You have to give them something of value or they will never come back. Ever.

If you can anticipate user questions and make sure that your site answers them, then you will be a successful Web publisher.

EXAMPLE 1: PERSONAL HOME PAGE

Suppose that you are building a humble personal home page. Why would a user come to your page? Someone might be trying to contact you or send you a package. So you obviously need to include your phone number, mailing address, and, perhaps, fax number. Perhaps you've invited some people to your house for a party and they don't know how to get there; they can't call you because you're out shopping for bagels. So you should have a map to your house (I just made my home address a hyperlink to the Yahoo map server).

Imagine that a friend of yours is at a party talking to Dale, a beautiful member of whatever sex you happen to fancy. Your friend is describing your charity, great humanity, and kindness to animals. Dale, however, won't agree to meet you without seeing a photo first. So you'd better hope that there is a portrait of you somewhere on your home page before Dale and your friend stroll over to the WebTV.

Or imagine that you've given an interview to a reporter on deadline. It is 2 a.m. when the reporter realizes he forgot to ask you for some background biographical information. You'll be getting a wake-up call unless you remembered to put a copy of your resume online.

At this point, some people might object that this information is too, well, *personal* for a personal home page. The Internet frightens them. Sure, their phone number is listed, and the price they paid for their house is public information. And their credit record is open to almost anyone who cares. But they think that if their name is known to an Internet search engine, suddenly all of the privacy that they supposedly formerly had will evaporate. Well, my home phone number has been available via the Internet for 20 years, and my Web site gets about 500,000 hits a day. I include a picture of myself naked with my old dog George and maps to my house and office on my site. And just as I typed that last period, sitting in what I thought was the privacy of my own home, someone I didn't know called me up. At 8:20 p.m. on a Monday. Would I make a donation to the March of Dimes?

Probably about 1 percent of my unsolicited phone calls are from readers of my Web site. Surprisingly enough, people who find me on the Internet seem to send e-mail instead. Go figure.

EXAMPLE 2: CAMERA MANUFACTURER

Suppose instead that you make cameras for consumers. Why would a user come to your page? Possibly to look at brochures, so it might be nice to have some advertising literature on your site. But the most important users are existing customers, such as someone who is traveling with a fancy Nikon single-lens reflex and has forgotten how to work the flash. Caught without the owner's manual, this customer has surfed in from an Internet café hoping to find the full text online.

Someone with a broken camera will want to know how to get it fixed and how much it will cost. So you'll want to put up maps to your service centers, warranty details, and prices for out-of-warranty repairs. Your site is a natural focal point for customers to meet, interact, and share their experiences with each other. So you should have moderated Q&A forums, bulletin boards, and classified ads where your customers can contribute.

EXAMPLE 3: SOFTWARE COMPANY

Everything that I said about the camera manufacturer goes for the software company as well. However, the camera maker is under no obligation to demonstrate an understanding of the Internet. But if you are in the computer business, you can't afford to have your customers say "My 10-year-old built a better Web site than that last week."

All of your documentation must be online and current. Demonstration programs, ideally running on the server or as Java applets, should be available.

Customers ought to be able to purchase and download everything that you sell. If you can't use the Internet effectively, then why would anyone believe that you are capable of doing anything with computers effectively?

HOW DO YOU KNOW WHEN YOU ARE DONE?

You can objectively measure how well your site meets user needs. Are you getting a lot of e-mail questions? As soon as two users ask the same question, you should beef up your site to answer it unattended: Build a page, fix the navigation model, or tune your search engine.

Since you are a competent webmaster, you are running a full-text search engine on your site. Why not hack the CGI scripts so that the search engine sends you e-mail when a query results in zero matches. Given information about which user queries are failing, you can add content or keywords as appropriate.

At this point you can relax. You aren't a loser with a big budget, a lot of ugly graphics, and no traffic. Users are not leaving your site in frustration, shaking their heads, and saying "They just don't get it." Now that you are safe from the wags at suck.com, what can you do to make your site fulfill its potential?

BECOME ILLITERATE (THAT IS, PRESENT MULTIPLE VIEWS)

Look up "Bomarzo" in the Michelin Green Guide to Italy: "Extending below the town is the park of the 16th-century Villa Orsini (Parco dei Mostri) which is a Mannerist creation with a series of fantastically shaped sculptures." Compare that description to the photos in Plates B and C (in the color plate section of this book), showing just a tiny portion of the Parco dei Mostri ("Park of Monsters"). Do these not suggest a somewhat richer place than the sentence in the Michelin guide? Yet the sight of tourists slavishly following the Michelin guide is a commonplace. Something really fascinating and unexpected is happening in front of them, but they have their noses buried in the guide, trying to figure out what the next official point of interest is. Such a tourist is *literate*. Not literate in the "I read classics at Oxford" sense, but literate in the "knowledge is closed" sense. Everything about Italy can fit into a book. Perhaps the 350 pages of the Green Guide aren't enough, but some quantity of writers and pages would suffice to encapsulate everything worth knowing about Italy.

Oral cultures do not share this belief. Knowledge is open-ended. People may hold differing opinions without one person being wrong. There is not necessarily one truth; there may be many truths. Though he didn't grow up in an oral culture, Shakespeare knew this. Watch *Troilus and Cressida* and its five

perspectives on the nature of a woman's love. Try to figure out which perspective Shakespeare thinks is correct.

Feminists, chauvinists, warmongers, pacifists, Jew-haters, inclusivists, cautious people, heedless people, misers, doctors, medical malpractice lawyers, atheists, and the pious are all able to quote Shakespeare in support of their beliefs. That's because Shakespeare uses the multiple characters in each of his plays to show his culture's multiple truths.

In the 400 years since Shakespeare we've become much more literate. There is usually one dominant truth. Sometimes this is because we've truly figured something out. It is kind of tough to argue that a physics textbook on Newtonian mechanics should be an open-ended discussion. Yet even in the natural sciences, one can find many examples where the culture of literacy distorts discourse.

If you were able to stay awake long enough to read through an academic journal for taxonomic botanists, you'd learn that not all botanists agree on whether Specimen 947 collected from a particular field in Montana is a member of species X or species Y. But you'd see quite clearly that everyone publishing in the journal agreed on the taxonomy, or how to build a categorization tree for the various species.

Note: I learned all of this interesting stuff about taxonomic botanists from Peter Nürnberg; see his paper at <ftp://bush.cs.tamu.edu/pub/publications/papers/96.webnet/96.webnet.html>.

However, if you were able to stay awake long enough to get through a cocktail party in a university's department of botany, you'd discover that even this agreement is illusory. There is widespread disagreement on what is the correct taxonomy. Hardly anyone believes that the taxonomy used in journals is correct but they have to stick with it for publication because otherwise older journal articles would be rendered incomprehensible. Taxonomic botany based on an oral culture or a computer system capable of showing multiple views would look completely different.

Open today's *New York Times*. A Republican politician is arguing for relaxed regulations on widgets. A Democrat is quoted arguing in favor of tightened regulations. There is a vote; widget regulations are tightened. On the surface, it looks like multiple perspectives. Yet at a cocktail party, your friend Sue might argue that the government shouldn't be regulating widgets at all. Joe would probably interrupt her to say that widget regulation is a canard; we really ought to talk about flag burning. Dana would bring up the *Simpsons* episode where Bart went on a school field trip to the widget factory. Alan would tell his grandfather's widget factory stories from World War II. Elizabeth would talk

about how she was surprised to see that they had no widgets in New Zealand and apparently did not miss them. Compared to the texture of the cocktail party, the *New York Times* article sounds like two rich old white guys saying more or less the same thing.

Some people like a one-truth world. If you have a huge advertising and PR budget, then you can control your public image very effectively in a literate world. Ford Motor Company has enough money to remind you 2,000 times a year that "Quality is Job One;" unless you lost a friend in a Pinto gas tank explosion, you probably will eventually come to agree. Microsoft via the genius of Bill Gates invented the mouse-windows user interface, reliable operating systems, affordable computing, and the Internet; if you don't think all that is true, ask someone who has never used a computer and whose only exposure to the industry is through mass media.

Perhaps it is because I'm a few billion dollars short of the necessary funds to create a one-truth world of my own, but I think the greatest artistic achievements hold the mirror up to a multiple-truth life. The Internet and computers, used competently and creatively, make it much easier and cheaper to collect and present multiple truths than in the old world of print, telephone, and snail mail. Multiple-truth Web sites are much more interesting than single-truth Web sites and therefore will get a lot more traffic. For example, the car manufacturers' sites are mostly collections of product brochures tarted up with flashing graphics. They get minimal traffic compared to the plain text rec.auto.* newsgroups, which present the real experience of car owners from around the world. The newsgroups don't have pictures, animation, sound, or video clips. But they have multiple truths.

OK, enough philosophy and Shakespeare. This is the point in the infomercial where the guy wearing the CAT Diesel cap asks, "Do I need a college education to build one of these here multiple-truth Web sites?" The answer is no.

- *Step 1: Put up some magnet content.* Your distribution cost is now zero. So take every document that you've ever distributed to anyone and put it on your site. In my case, I'll write a 50-page story about Costa Rica and illustrate it with 200 pictures (see Plate D) before inviting people to contribute their wisdom.

General Electric (<http://www.ge.com>) is a rare good example of a corporate site that takes the Web as distribution medium seriously. They have brochures, yes, but also owner's manuals and installation guides for every GE product. If you moved into an apartment with a GE appliance but the previous tenants did not leave you the instructions, you can grab a PDF file from the GE site and print it.

- ▶ *Step 2: Develop technical means for collaboration.* Your Web server should indeed be barfing out your content at a steady clip, but in a way its most important role is to slurp up content from users. My photography sub-site (<http://photo.net/photo>) pumps out 400,000 files every day with my photographs and writing. More interesting to me, though, are the comments collected by my Loquacious system for each page and my indexed categorized question-and-answer forum, shown in Figure 1.1. Users are helping me build a great repository of photography knowledge. I'm pretty diligent about responding to questions in the forum, but my happiest hours are spent watching users answer each others' questions. You don't have to be a traditional publisher per se to benefit from collaboration tools. Manufacturers can collect and redistribute consumer comments about dealers who carry their products. Academic researchers can collect comments, ideas, and questions sparked by their writings. An example of such responses can be seen in Figure 1.2.
- ▶ *Step 3: Be prepared to interact with users.* This means that if you have a question-and-answer forum, spend the time to answer the questions. It also means moderating the forum to delete uninteresting threads. It means killing idiotic comments on the comment server but keeping the relevant ones, even those that savagely disagree with your point of view.

Does this sound like too much work? If your site is commercial and you claim to be a smart businessperson, then you ought to be able to figure out a way to interact with users more cheaply over the Internet than via 800 numbers. If you are a writer or a photographer who has built a non-commercial site, then disintermediating the publisher means total artistic freedom. User feedback might be annoying at times, but if you want to be insulated from your readers, why publish on the Web at all?

THINK OF THE WEB AS PRIMARY

A site that exists primarily to tease people off the Internet into buying something will almost always cry out to be ignored. Web sites should stand alone.

Joe Greedy puts up a site showing the cover of his book and a headline: "Buy me for \$17." You'd think that this would be the lamest possible Web site. But Mr. Greedy manages to earn extra suck points by making sure that the only ordering option is by telephone. That way the modem crowd will be forced to write down the number on a Post-It and then disconnect before they can order their pile of processed tree carcass.

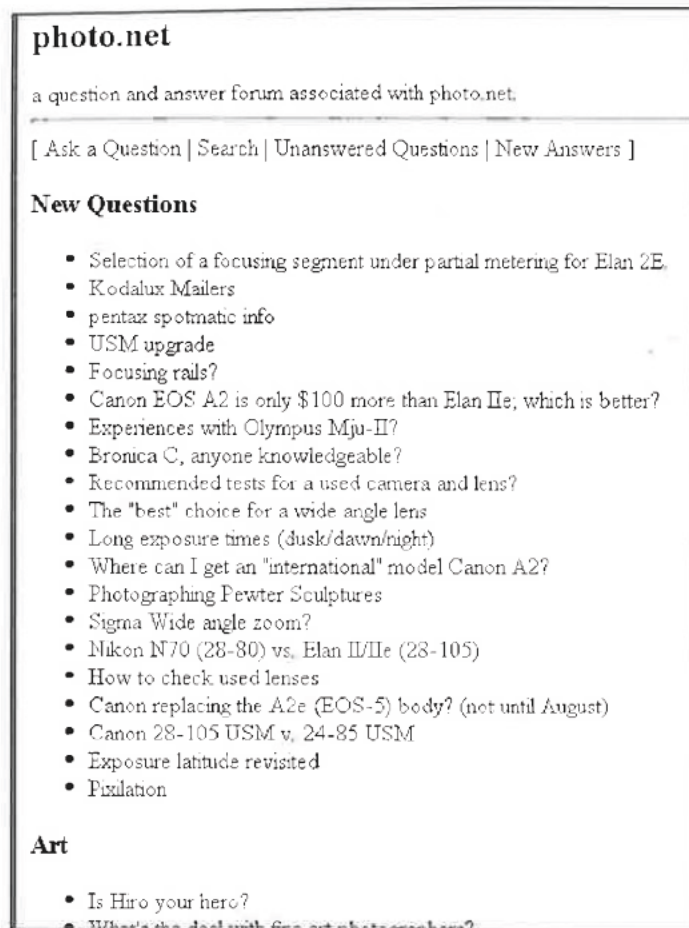


Figure 1.1: The question-and-answer forum in <http://photo.net/photo>, which contains more than 1,500 archived messages. Answers are e-mailed automatically to the user who posted the question, even months later. Users can choose to be notified by e-mail of new messages instantly, daily, or weekly. (Note: If you'd like to run a similar forum on your own site, just visit <http://webtools.com/wtr/> and fill out a form. The software is free and you can even use one of my Web/RDBMS servers for free.)

Jane Clever puts the full text of her book online. With several hundred pages of text instead of one, her content will be several hundred times more likely to attract users of search engines. By enabling online ordering for those dead-tree huggers, Ms. Clever will sell far more copies of her book than Mr. Greedy. Also, with 100,000 hits per day, Ms. Clever can sell ads on her site and links to other Web publishers.

Another kind of site that almost always disappoints is the adjunct to the physical event. Pathfinder did a companion to a traveling museum show by

Comments

on Terezin Concentration Camp (made possible by Loquacious)

In 1995 I visited Dachau Concentration Camp, just outside Munich, Germany. My feelings are similar to those of the author, I agree that EVERY person should visit a camp if at all possible. It's an unforgettable experience, I was not so upset emotionally as one might be as I was prepared by my visit to the American Holocaust Museum (correct name?) in Washington, D.C. It's an outstanding museum, one of the world's best. It gives one a great overview of the entire holocaust and will certainly contribute to one's understanding at camp visit.

Contributed by Pete Wenzel (pw@gwcc.com) on January 20, 1997.

At a music camp that I went to I sang a song about Terezin. I discovered that one of the saddest things about this place was the fact that a lot of the children tried to keep their hope alive through art. The song that we sang was about the ghetto at Terezin and how the children would hear the soldiers marching footsteps. These poor children had only one hope, and from what I discovered, they really did try to hold on to even that small thing when everything else was gone. Extremely Sad!

anonymously contributed on February 08, 1997.

Add a Comment

philg@mit.edu

admin page

Figure 1.2: The Loquacious comment server collecting reader responses to my Berlin/Prague story (<http://webtravel.org/bp/>). The original 30-page story and 80 photographs attracted a reader-contributed story about the Terezin concentration camp. That story in turn attracted the comments that you see here (<http://db.photo.net/com/philg/bp/terezin.html>). If you'd like to use the Loquacious system for your own Web site, just visit <http://webtools.com/wtr/>. It is free.

well-known photographers such as Annie Liebowitz. The biggest available pictures on the Web site were tiny, occupying about 1/100th of my 20-inch screen. The editor of the site asked me to add a link from photo.net to the Pathfinder site. I refused, saying that my readers were accustomed to filling their screens with the 500x750 pixel and 1000x1500 pixel images on my site. They would just get frustrated squinting at the tiny Pathfinder images. He replied that the museum show photographers were concerned about copyright infringement. I said "Well, maybe they shouldn't be on the Net. They are getting plenty of promotion in bookstores, museums, magazines. Why don't you find some less known photographers for whom the Web is primary?"

In October 1995, the MIT Media Lab threw itself a tenth anniversary party. I skipped it so that I could take snapshots of colored trees in Vermont, but my friends said that the physical event was fabulous: little gifts for everyone, cleverly packaged; a Photomosaic poster by Rob Silvers; women in plastic pants. As an adjunct to the physical event, the Media Lab was going to create the best Web site ever. They got NYNEX to bring in a 45-Mbps T3 network connection; you wouldn't want millions of users to get slowed down working their way through MIT's 100-Mbps backbone. They got Hewlett-Packard to donate a huge pile of multiprocessor machines with disk arrays. They hired expert consultants to plug all the computers in and hook them up to the network. They hired professionals to do graphic design and site layout.

After getting back from foliage country, I visited <http://www.1010.org> with high hopes. There wasn't any magnet content. Nobody had bothered to write stories or take pictures. Every day a Media Lab editor posed a question and then sat back to watch a Usenet-style discussion evolve. There were only a handful of postings in each area. One user had contributed a smiley face. Colon dash right-paren. That was his entire message. This didn't really shock me until I noticed that on a scale from 1 to 7, this post had been rated 4.3 by other users. Yes, several other users had taken the trouble to rate this three-character posting. When the ten-day Web event was over, the massive disk arrays held almost enough user-contributed data to fill two 3.5 inch floppies.

How could so much money and hardware have resulted in a site that an elementary school would have been embarrassed to make public? The Web site wasn't primary. The party was the important thing. The Media Lab, one of the last groups at MIT to put up a Web site, had no real Internet culture. They wanted to reach corporate managers and bring them into the physical space of the Lab. Since corporate managers didn't tend to have a TCP/IP connection on their desktop, it wasn't obvious how the Internet would be useful in attaining the Media Lab's goals.

Note: Don't try to visit <http://www.1010.org>. The Media Lab eventually got Web-savvy enough to realize the embarrassment value of such a site and pulled the plug on the server.

Money is nice. Bandwidth is nice. Graphic design is occasionally nice. But your Web site will suck if you see it as a pimple on the butt of something much larger.

SUMMARY

The stuff in this chapter isn't meant to be true. I'm not a great writer like Shakespeare (yes, I know that I could write this observation up for *Duh* magazine; you don't have to send e-mail). I'm unable to transcend the medium of the printed page and present multiple truths. My hope is that I have provoked you into doing the following:

- ▶ Putting yourself into your user's shoes before planning your site's content. Why would they come to your site and what can you do to make sure that their visit is rewarded?
- ▶ Thinking seriously about the Web's potential as a medium for collaboration. In intelligent, thoughtful hands, the Internet is more powerful than the printed page.

This chapter was about people talking to computers and people talking to each other with a computer mediator. In the next chapter, we'll talk about how computers talking to each other could finally deliver all of the benefits that the Computer Age was supposed to have conferred on us (and, if you insist, how you can make a buck off of it).

So You Want to Join the World's Grubbiest Club: Internet Entrepreneurs

THE STEAM ENGINE AND THE RAILROAD

FOUR WAYS TO MAKE MONEY FROM
YOUR SITE

WE LOSE MONEY ON EVERY HIT BUT
MAKE IT UP ON VOLUME

A FINAL PLEA

Chapter

2

“**T**he guy with the Web site.” That’s how my friends introduce me at parties. I guess that says something about what a rich, multifaceted, textured personality I’ve developed by spending 15 out of my 33 years at MIT. Yeah. Anyway, the response from my new acquaintances is invariably the same: “How are you going to make money off your Web site?”

If they’d been told I’d spent \$20 on a subscription to *Playboy*, they wouldn’t ask how I intended to make money off that. If they knew I’d splurged for a \$5,000 Viking stove, they wouldn’t ask if I was going to start charging my brunch guests \$5 each. If I told them I dropped \$20,000 on a Dodge Caravan, they wouldn’t ask if I was going to charge my dog \$10 for every trip.

Web site hosting can cost less than any of these things; why does everyone assume that it has to make money? I did not set up my site to make money. I set up my site so that my friend Michael at Stanford could see the slides I took while driving from Boston to Alaska and back. It turns out that my site has not made money. Yet I do not consider my site a failure. My friend Michael at Stanford can look at my slides anytime he wants to by typing “<http://webtravel.org/samantha/>”.

I’m not saying that this should be everyone’s goal. After all, you might not know my friend Michael at Stanford. Or, if you do know him, you might not like him. But keep in mind that if you are destined to lose money on your site, it is much less humiliating if you can say that making money wasn’t the idea.

If you can’t remember that greed is one of the seven deadly sins, then I suppose it is worth thinking about how to make money on the Internet. First, ask yourself the question, “Why do we think that there *is* money to be made off

the Internet?" Karl Taylor Compton, former president of MIT, said it best in 1938: "In recent times, modern science has developed to give mankind, for the first time in the history of the human race, a way of securing a more abundant life which does not simply consist in taking away from someone else."

I believe that *computers* will secure a more abundant life for the human race. Mine would not have been a controversial statement in 1960, when IBM was just beginning to saturate corporate America with mainframes. It was obvious to everyone in 1960 that computers were going to usher in a new Age of Leisure. In 1997, my statement seems absurd. Computers have been around for fifty years now without doing much for the average person. In fact, it is a commonplace among economists that computers have reduced the productivity of American business.

Computers by themselves are a liability. Getting information into and out of them is so expensive that using paper and file cabinets is probably less trouble. Sure, in 1997 we don't see too many people manually writing payroll checks or calculating artillery shell trajectories. But most people get through most parts of their day without relying on a computer. Why don't I think that will be true in another 50 years? Because of the network.

THE STEAM ENGINE AND THE RAILROAD

James Watt's 1765 steam engine didn't change your life unless you were pumping water out of coal mines. A steam-powered factory could have produced enough goods to supply an entire nation, but since there was no way to distribute that output there wasn't much point in building such a factory. With the railroad came the ability to serve a national market with one factory. Factories grew enormous and pulled people out of the countryside to work inside the Satanic Mills. It was the mounting of the steam engine on rails and the spreading of railroads across nations that transformed society, not the steam engine *per se*. The computer is the steam engine of today; the network is the railroad.

With a ubiquitous network, all of the information that you consume will arrive in a machine-readable form. If the information is structured appropriately, as with standards like Electronic Document Interchange (EDI), it will arrive in a form that is immediately applicable to your internal databases without human intervention. Your computer will truly be able to handle routine transactions on your behalf.

When the network infrastructure is powerful enough that most houses enjoy video-rate bandwidth, the computer will be able to support your collaboration with other people. If you had TV-quality video and audio links to your collaborators and a shared workspace, you wouldn't have to commute to work

or fly around from city to city so much. Though I don't like to predict the demise of a 3,000-year-old trend toward urbanization, it indeed seems possible that collaboration tools might enable some people to move out to the country yet still keep their urban jobs.

The Internet will change society. Some people will get rich off that change. I guess you might as well be one of them. Probably your safest bet is to figure out where all those telecommuters are going to end up living and open a little mall there with a McDonalds, MicroCenter, and Trader Joe's. But this is a book about building publicly accessible Web sites, so here are my ideas.

FOUR WAYS TO MAKE MONEY FROM YOUR SITE

I think that all consumer-oriented Web services fall into at least one of the four categories below. If you are starting a new site for the purpose of making money, it is worth considering which categories your planned services fall into and how hard it is to make money in those categories.

Here are the categories:

- 1 *Sites that provide traditional information.* This type of site, shown in Figure 2.1, requires the least imagination but also the most capital investment. Find bodies of information that consumers in the 1980s bought offline and sell them online. This includes movies/videos/television, newspapers, magazines, weather reports, and stock market information. A classical example is an online magazine or newspaper. Revenue comes from advertising, links to sites that do retail transactions and give you a kickback, and occasionally subscriptions. Putting up a Category 1 site requires almost no imagination, technology, or investment, but it will cost you a fortune in the long run as you keep buying content to keep it fresh.
- 2 *Sites that provide collaboratively created information.* This is information that was virtually impossible to collect before the Internet. A dead-trees example would be the *Consumer Reports* annual survey of automobile reliability. They collect information from their readers via mail-in forms, collate the results, and publish them once a year. The Internet makes this kind of activity less costly for the provider and provides much more immediate and deeper information for the user. For example, each user can **tell you how frequently his car has broken down and you end up with a valuable database of car reliability statistics to give back to users** collectively. Revenue comes from the same sources as in Category 1 but production expenses are lower. It costs money to establish your server as a forum for users to exchange data. You have to buy some magnet content