



## The New Publishing

### An Information Choreographer's Manifesto

**Abstract:** We know we can do better than lowest-common-denominator HTML. "Better" means that it is easier to use, more compelling, "sexier", and more fun for us to design and produce. But it's very expensive to do bleeding edge for lots of content, and **too** expensive to do bleeding edge and a version that works in old browsers on slow modems on slow computers with small monitors.

The solution is to separate the content from the look and feel using templates, create the content with structure that can be manipulated by scripts, and to put the site structure in a database, separating it from the templates. Now you can automate to make site-maintenance easy, and you can easily generate multiple versions of the entire website. You get bleeding edge and lowest-common-denominator. You can also handle platform inconsistencies this way.

### Call me "Information Choreographer"

My business is scripting the flow of information: creation, maintenance, generating versions of the site, getting the right files onto the server, delivering the right versions of files based on user and display device.

This is not your "church website".

As soon as you have two people working on a website, you need source control (or a circus clown).

I can create Word templates and scripts that give you most of the power of XML inside the word processing software almost everybody has to use. People who write can create and maintain the content in Word, without knowing a lick of HTML. My software generates the HTML that other scripts use as input to create the website.

The structure of the documents allows you to target content for different purposes and different types of users. Specifically, you maintain one source file, but the content can show up in different ways in different places, or for different users. For example, in online help you can have slightly different content for context-sensitive help, versus the regular documentation. For that matter, you can have special versions for advanced and beginning users.

The cost of maintaining the website falls dramatically, you have time to focus on quality, and you get richer content and display. Plus, you get the fun of creating the bleeding edge version without having to compromise. (Slow modems get the basic version.)



## The Revolution

The revolution is over! Publishing will never be the same. This is a post hoc manifesto: what you have to do after the revolution, rather than what you have to do to accomplish the revolution.

The revolution was control of the display device.

There was speech. Then came story. We got writing. We got the printing press. A few years ago we got desktop publishing. Always, the "display device" (the paper it was printed on) was controlled by the people distributing information.

Not any more.

Now you distribute the information to someone else's device: computer, monitor, over their limited bandwidth.

We've just lived through a temporary period in which you could make sure your site works on a couple of different browsers, and say "to hell with" any potential customers who don't have one of those browsers.

That time is over, too. If you're going to sell over the web, there's no more "screw the user". If you're doing branding, same thing.

Unless your site is just a dumb marketing site — if there's any reason at all that people need or want information or transactions from you — your valuable customers will come to your site with browsers built into cell phones, televisions, PDAs, and before long from toasters. And even if most people have fast connections, some will have slow connections (calling from Prague), or be stuck behind overloaded firewalls (where one big file is fast, but twenty small files are excruciatingly slow).

There's another difference, that is one of degree (orders of magnitude).

In the old world, you would print a book, and that would be the end. If you wanted to print it again, you might have had to typeset it again.

The new world is "never-ending" publishing, in multiple forms. A stinking press release gets printed out to fax, gets sent by email, could even be made into a .pdf file, and goes onto the website in HTML. Every time your website look-and-feel changes, you have to upgrade your whole archive of press releases.

It isn't just press releases... it's everything.

The only reasonable way to deal with this is to use a structured template to create the press release (and other files) in such a way that all the other versions of it are produced automatically.



Your archive of content should be the original source versions, not the HTML output files. Whenever the look-and-feel changes, you run the new templates on the old files and you're done.

## The Manifesto

The web is not a Macintosh. Desktop publishing on the Mac enabled us to tweak every piece of content, to get perfect layouts. You can no longer afford to tweak each piece of content every time it gets used. No matter how hard you tweak, your files will look different on different display devices. Unless the lowest-common-denominator is your aspiration, this means two or more versions of your site. The only choice is to automate.

To automate, you must create your content using structured templates (like XML). The templates must be well thought-out, so you can generate all the output forms you will need directly from the source files — without having to tweak each piece of content.

You must keep the structure of your site in a database. You will need to generate different types of navigation for the different versions. If you put the navigation in your templates, you will forever tweak. You will need boatloads of galley-slaves to do your HTML.

If your website is trivial and you have deep pockets — or are willing to make your customers wait — you can do all of the processing on the server whenever someone clicks on a link.

If your website is complex, you will want to do processing on files before they go to the server. This way, the server doesn't have to parse every byte of every file, it just chunks the flat file out to the browser. Flat HTML is fast and reliable. Scalable.

Database folks will whine. "But whenever you change the template, you have to update all the files on the server." This is the information age. If you can hit one key and generate 10,000 flat files, you are way ahead of forcing the server to build each of 500 files every time a user requests it. With source control, automation, and scripts to check for invalid html and missing files, you can verify that the files are right, no matter how many files there are.

This leaves your server free to do the things it should do: decide **which** flat file to send, based on the browser type, the user type, the size of the monitor, or any other relevant considerations. Doing this for the content sections of your website frees up processing resources for the transaction sections.

## Nirvana (Grows Corn)

The result is that you get to make the bleeding-edge website that we all want to work on, that we know works better for users, that we know works better for branding and transactions. It's even cheaper. You're beyond cutting and pasting into templates.



## The New Publishing

You're beyond creating the parallel navigation for every page with the "copy-paste-tweak" cycle.

You don't have to exclude potential customers to do it.

Suddenly, the dream is reality. Our websites are aligned with our values. People on slow modems and old computers in central Africa can use your website to improve the quality of their lives.

When the publishing is easy, you have energy to check for quality, and to create the amazement that we all deserve. Mediocrity is for the old world.