

How this journal is made

Jaromil's Journal of Musings

February 26, 2010

0.1 About the technology of this journal

The GNU Emacs¹ editor originally written by Richard Stallman² is used for writing, archival and publishing of all documents presented in this journal: thanks to the Muse Elisp extension³ all pages (edited locally in simple text) are automatically rendered as a hyper-linked website as well as PDF files with a fine typeset.

This setup shifts most CMS⁴ functionalities to the client-side, while removing the requirement for a big web server and database: all documents can be edited and previewed while off-line, the final content can be served from a battery operated business-card sized⁵ web server⁶ taped behind the toilet of a cyber-cafe ;)

In fact, publication of static html pages relies on a simple web space instead of depending - and often licensing to - a third-party "content provider".

My personal opinion about Web2.0 is mostly unprintable⁷ and with this journal I'm happy to demonstrate an efficient alternative to its centralised model of content provision, stepping back from the freedom that is offered by the Internet.

Still someone reached to print out these opinions in a presentable way: let me follow with a text taken from the concepts behind webmake⁸, a web-page template engine written in Perl language.

As consciousness have progressed on these issues, there is a remarkable amount of new projects implementing static generation of websites, among them: MovableType⁹, StaticMatic¹⁰, Jekyll¹¹ and others...

The Web Is "Read-Mostly": Bake, Don't Fry

by Justin Mason

Several other similar web site management systems revolve around dynamic code running on the web server, which assembles the pages as they're requested by the client. In the terminology used by Ian Kallen when building Salon.Com¹², they "fry" the pages on-demand.

For most sites, the pages do not change based on which client is accessing them, or if they do, they don't change entirely; perhaps an extra set of links becomes available in the page footer allowing a logged-in user to make modifications using CGI, or PHP or Perl code, but that would be it. The page just isn't volatile enough to require continual re-generation for each request.

¹<http://www.gnu.org/software/emacs>

²[http://en.wikipedia.org/wiki/Special:Search?search=Richard Stallman](http://en.wikipedia.org/wiki/Special:Search?search=Richard+Stallman)

³<http://www.mwolson.org/projects/EmacsMuse.html>

⁴<http://en.wikipedia.org/wiki/Special:Search?search=CMS>

⁵<http://hackaday.com/2008/09/18/web-server-on-a-business-card-part-1/>

⁶<http://hackaday.com/2008/09/25/web-server-on-a-business-card-part-2/>

⁷[piracy_privacy.html](#)

⁸<http://webmake.taint.org/doc/concepts.html>

⁹<http://www.movabletype.org/>

¹⁰<http://staticmatic.rubyforge.org/>

¹¹<http://github.com/mojombo/jekyll>

¹²<http://www.arachna.com/edu/talks/iswp/index.html>

As a result, all this churning about, generating pages on the fly from its raw components each time, is wasted; it just eats the server's CPU and memory for no real gain, and introduces yet another breakage point (databases, memory usage, the /. effect...) where things can go wrong, just when you're not looking at it.

WebMake takes the "baking" approach, generating virtually all its output before the web server gets involved. The web site admin runs the webmake command, and this generates the pages.

Note that WebMake doesn't preclude dynamic content in the pages, however. PHP, CGI, ASP or embedded Perl code can be used, and WebMake will not interfere. In fact, a future version of WebMake will probably provide some "fried" features of its own...

About Multi-media content

All audio and Video published here is encoded using **OGG** technology, an open and free format to compress audio and video : free from any patents, Vorbis and Theora are efficient alternatives to MP3 and MP4, **they sound good and they feel even better** ;) I advocate use of Ogg files as a standard for multimedia encoding, so there can be a format that anyone is free to implement and use into free software and open hardware players. For more informations see the PlayOgg campaign¹³.

Copyright (C) 2000 - 2010 dyne.org foundation and respective authors. Verbatim copying and distribution is permitted in any medium, provided this notice is preserved. Send inquiries & questions to dyne.org hackers.

¹³<http://www.fsf.org/resources/formats/playogg>