

Cell: +1.512.784.7526

Home: +1.512.346.5240

Office: +1.512.788.5254

sauer@technologists.com

7123 Valburn Drive

Austin, Texas 78731

<https://technologists.com/~sauer>

<https://vidconf.net/>

Facebook/Google/LinkedIn/Twitter: CharlesHSauer

[eight Jurassic O.S. on 1992 Dell 486D/50](#)

[Reviving timbl’s WorldWideWeb browser](#)

[Sustaining Dell UNIX](#)

MW site unification – PET International rebranded itself as Mobility Worldwide in 2016. At the same time, I transformed the majority of the 20+ Affiliate Web sites to become components of MobilityWorldwide.org and transformed the site to make the unified site adaptive to, and optimized for, mobile devices and portrait mode screens.

WordPress content management (2010): The PET Project

(<https://petinternational.org/>) had been struggling for a few years with the need for updating, coordinating and consolidating web sites. A Joomla!-based portal proposal was meeting resistance due to cost and other considerations. I developed and gained acceptance of an alternate, nearly free, approach based on using WordPress for content management. That approach is now used for <https://petinternational.org/> and an increasing number of affiliate sites. In the process, I became more fluent in WordPress in general, and gained understanding of tradeoffs between WordPress.com and WordPress.org, of limitations of various WordPress themes, and of approaches to modifying PHP and CSS files to achieve desired results.

Revised appearance modeling macros (2007): With a project to mimic a paper brochure with web pages, it was time to revisit CSS, potentially abandoning the appearance macros I had developed in 1997 and used for a decade. It turned out that the macros still provide a useful role as long as CSS is used for the things for which CSS is intended. See [CSS: "A man's got to know his limitations"](#).

“Good News Connection” (2006): GNC was designed to facilitate typical church based administrative tasks by providing web forms and procedures. GNC was implemented in PHP. Though technically successful, the approach was not compelling enough to displace traditional, paper-based approaches. (See [Good News Connection](#).)

Porting/Testing SITEFORUM for OS X Server (2004-2005): In 2004, SITEFORUM intended to provide Java-based web portals hosted on a variety of platforms. It was believed that in education markets there was good opportunity for

SITEFORUM on OS X Server. I developed installation procedures and scripts sufficient to install SITEFORUM as a standard DMG, which adapted SITEFORUM to platform differences, e.g., the use of Apache 1.3 vs. Apache 2, and handled issues such as DNS for a non-Internet environment in a classroom.

Lighting tunneling prototype (2001-2002): The Lighting architecture was designed to give a mobile device, e.g., the Palm OS smartphones available at that time, remote control of the desktop and a secure telescopic extension of the desktop rather than being a client to specialized “wireless” servers. It was believed that Java would be a portable environment across mobile devices. I designed and partially prototyped desktop “agents” in Java that would act on behalf of the mobile device, authentication servers for the mobile devices, and TLS based tunneling software to provide encryption between the mobile device and the desktop.

Cymbia Telecom Analyzer (1999-2001): After *Recruiter*, I was eager to try other languages and technologies in a new startup, Cymbia, (<http://web.archive.org/web/20000829080322/www.cymbia.com/about.asp>), particularly Active Server Pages, Javascript and Java. One of the principle features of *Telecom Analyzer* (<http://web.archive.org/web/20000829080330/www.cymbia.com/products.asp>) was extensive browser-based report generation and customization. These reports were generated with Javascript after *synchronously* fetching XML data from the (SQL) server – XMLHttpRequest was *not* used. In 2001, our seed investment funds were exhausted and sales revenue was insufficient to pay salaries. I became the sole developer, and continued to enhance and maintain all the code, including the Java applet & servlet.

Appearance modeling macros for *Recruiter* (1997-98): It rapidly became apparent that installing *Recruiter* on customers’ servers was not going to be successful for a variety of reasons including unique technical challenges and need for I.T. buy-in and cooperation. It was also evident that the appearance of *Recruiter* pages needed to match the other pages on the customers’ site. CSS and content management systems were new and unproven. I decided to try to re-cast *Recruiter* pages using [m4](#) macros that would facilitate a broad variety of appearances. A person with sufficient skill could, in a matter of hours, generate a *Recruiter* instance closely matching a customer’s site in layout, fonts, colors, images, etc. This “appearance modeling” was very successful with customers, developers, management and prospective investors. The *Recruiter* page at <http://web.archive.org/web/19990125090055/http://hire.com/> was generated using these macros, as were all of the customer sites given as examples on that page. (Unfortunately, most of those links are not available from web.archive.org, but at this 2019 writing the archived CareerStop.org, CORESTAFF.com, and EDS.com links work to some extent.) Another way to think of these macros is that they comprise a static content management system – the content is stored in m4 files, which are transformed into HTML in advance, vs. more dynamic page generation in a typical content management system. I have continued to

use these macros on my own and other sites, gradually adapting them to expect and take advantage of CSS. For example, https://technologists.com/sauer/chs_bio.m4 processed by m4 produces https://technologists.com/sauer/chs_bio.html.

Productization of world.hire Recruiter (1996-97): *Recruiter* was in prototype form, based on PERL CGI and [mSQL](#), when I became the first non-founder employee at **world.hire**. It was intended that *Recruiter* be hosted on the customers’ servers. I personally installed and supported the initial customer sites, hosted on Apache on Linux and Solaris. Subsequently, I built and supervised the team that supported the Apache mSQL version and also created a version suitable for hosting on IIS 2 and SQL Server on Windows NT Server 4.0.

VTEL Labs Intranet (1996): An early system for tracking and reporting research activities, hosted on IIS 1 on Windows NT Server 3.5.1.

Single System Image Daemon (1987-88): A major feature of AIX Distributed Services was the presentation of a single file system hierarchy across participating machines, using fine granularity mounts. See [Presenting a Single System Image with Fine Granularity Mounts](#). I created a daemon for managing dynamic mounting and unmounting of individual files.

PL.8 to C translator (1986-87): A major portion of [AIX](#) 1 & 2 device drivers, memory management and other kernel-like components were written in [PL.8](#). An objective for AIX 3 was to eliminate PL.8 in favor of C. I prototyped a PL.8 to C translator in [Icon](#). I then supervised the production of a more robust translator which was used extensively in AIX 3 development.

RESQ (1975-1984): The Research Queueing Package (RESQ) was one of the earliest and most influential packages of queueing network software. See [Performance History - RESQ](#) for a synopsis and [The Evolution of the Research Queueing Package](#) for a more thorough overview. My contributions included design and definition of “extended” queueing networks, architecture of RESQ as a whole, and personal responsibility for almost all of the simulation components. At one critical early stage in the development, when it became clear that legacy use of Fortran for the simulation components was a major impediment, I created a crude Fortran to PL/I translator in SNOBOL. “The elapsed time between beginning work on the translator and getting a running PL/I version of APLOMB [the simulation subsystem] was approximately two weeks, and this achievement was a great relief to those who anticipated a much, much larger effort.” [Evolution, p. 6]. RESQ source code was made widely available to users within IBM. Based on that source code, a RESQ user I never met told a mutual friend at TANDEM that I was “the best PL/I programmer in IBM”.