

STEVE A FINK

sphink@gmail.com

OBJECTIVE

To develop cutting-edge, marketable products with a fun and highly motivated team.

SUMMARY

- Software architect with 10 years experience, including complete lifecycle of multiple systems.
- Experienced with all aspects of complex application creation: 2d and 3d graphics rendering, computer vision, gesture processing, monitoring, system management, training, and installation
- Excellent collaborator between different teams; often served as the primary "go to guy"

AREAS OF EXPERTISE

- **Software System Architect:** Key player in architecture, design, implementation, and field monitoring of a 300KLOC application and support structure -- software/hardware mix, more than 200 field deployments, remote monitoring and control, and content development and management. Author of a terabyte-scale text and code fuzzy matching engine and graphical user interface.
- **Tools Developer:** Creator of numerous support tools, tutorials, and documentation. Implemented remote debuggers, memory trackers, visual layout tools, log and trace analysis tools, QA automation harnesses, simulation frameworks, format importers and exporters, protocol/event model bridges, documentation processors, wiki automation, system and library call tracers, and many more.
- **Software Integrator:** Successfully integrated scripting languages, physics simulators, Flash, Java, web browsers, text editors, 3d modeling packages, XMPP (Jabber), and other third-party libraries and applications into a whole that makes available the strengths of the various constituent pieces without becoming an uncoordinated mess or requiring a specific toolset.
- **Open Source Developer:** Long-time member of the Open Source community, using and contributing to projects both professionally and personally. Submitted patches to numerous projects (see <http://0xdeadbeef.net/~sfink/hacking.html> for partial list.) Release pumping (manager) for early versions of the Parrot project. Speaker on Open Source topics at user groups and conferences.

WORK EXPERIENCE

Senior Software Architect, Reactrix Systems (Redwood City, CA). 2002-2008.

Reactrix produced a real-time, camera based interactive display used mostly for advertising and trade show events. The vision processing and graphics rendering system (called the "Effects Engine") is written in C++. Artists, interaction designers, and content programmers produced content in the form of a master XML file referencing various images, animations, 3d models, scripts, etc. that collectively defined the visual appearance and the user interaction. The system is comprised of an embedded Linux box, an LCD panel or projector, and an infrared 2d or 3d camera.

- Designed and implemented* flexible engine for interactive vision-driven simulation and rendering
- Embedded scripting language for high-level control of Effects Engine
- Conceived of and wrote popular newsletter to communicate engineering topics to creative department
- Created standalone laptop version of system (sales demo unit)
- Integrated alternative Adobe Flash front-end for animation and rendering

- Initial version of content release and management system
- Debugging tools: remote dataflow debugger, script debugger integration, data injection tool, memory tracker, vision/camera input replay
- Customized continuous integration tool (converted from email to XMPP, added packaging)
- Diagnosed field issues with Network Operations division, automated error detection and recovery
- Lightwave3D model import and animation*
- OpenGL save-to-stream tool used for client approval process and QA automation
- OpenGL shader development and templating
- Flexible test harness and output analyzer
- Wiki installation*, maintenance, extensive customization
- For more, see <http://0xdeadbeef.net/~sfink/resume/Reactrix-xp.html>

*Done jointly alongside other team members; all other contributions were accomplished independently

Consultant, *iAuctionNet (Tokyo, Japan). 2002.*

"Boutique" online auction for the Japanese collectible market. I spent two months living in Tokyo and working on the auction just before its launch.

- Designed, and implemented system for automated scanning and statistical analysis of competing Japanese auction sites
- Installed Oracle database, Orion J2EE application server, bug tracker, and VPN.
- Handled internationalization issues.

Founding Engineer, *Digital Integrity (San Mateo, CA). 1998-2002.*

Digital Integrity detected similarities between documents to detect duplication (usually looking for plagiarism, code theft, or reuse.) The technology applied to code as well as English text, and would work on large repositories (eg 1 sec response on a 3TB web search).

- Implemented linear-time algorithm for finding content matches in large repositories
- Managed 3 engineers
- Created web application for finding, displaying, and analyzing matching documents
- Wrote clustering and checkpointing software to utilize 24 commodity Linux workstations
- Co-designed and implemented side-by-side document similarity visualizer based on heuristic suffix-tree traversal algorithm
- Implemented email firewall that blocks messages containing sensitive information
- Constructed cross-platform, cross-language RPC IDL parser and code generator
- Wrote utility for automated SQL schema migration
- Co-developed significant improvements to core algorithm (issued patent #6757675)

Open source. 1996-present.

- *flick*, a utility that uses the Flickr web API to upload and download photos, detect and delete duplicates, and perform scripted batch operations
- Regular expression compiler for the Parrot virtual machine underlying the upcoming Perl6.
- Regression testing and web automation scripting platform.
- *ucalc*, a calculator for performance estimation and unit-carrying calculations.
- Perl language plugin for source code printing utility.
- Visual debugger for LR-family parser generators, presented at 4th Perl Conference.

- Yacc-based parser generator that generates perl code.
- Linux kernel patch to add detailed open file information to /proc filesystem.
- Netscape browser plugin for executing arbitrary programs on input data.
- MMX assembly code algorithm for efficiently converting image data streams.
- Added support for arrays, structures, and typedefs to ltrace.

PATENTS AND PRESENTATIONS

- US Patent #7428542: *Method and system for combining nodes into a mega-node*
- US Patent #6757675: *Method and apparatus for indexing document content and content comparison with World Wide Web search service*
- Talk: Using Inline::CPP in Production Environment at San Francisco Perl Mongers (Nov 2006)
- Conference session: Visual Debugger for LR Grammar Parsers at O'Reilly Perl Conference (Jul 2000)

EDUCATION

Carnegie Mellon University, Pittsburgh, PA
Bachelor of Science in Psychology and Math/Computer Science, 1995.
GPA: 3.8

University of California at Berkeley, Berkeley, CA
Master of Science in Computer Science, 1998.
GPA: 3.9

COMPUTER SKILLS

Languages: C++, C, Perl, Java, Python, PHP, XML, XSLT, Tcl/Tk, SML, Assembler (MIPS, 68000, 80x86, ia64, others), Pascal.

Platforms: Linux (16 years), Unix (17 years), Windows (scattered).

Tools: Emacs, strace, ltrace, gdb, git, Perforce, Subversion, Bugzilla, Tinderbox, R, MySQL, SSH, iptables, nc (NetCat), Apache, mod_perl, bugle, GIMP, Lightwave3D, FFmpeg, ALSA, JACK
see more complete, categorized listing with full descriptions at <http://0xdeadbeef.net/~sfink/resume/tools.html>

(resume version sa1.4)