

David D Favro

✉ david@cv.favro.net (*preferred*) ☎ +1.248.284.2444 🌐 <https://cv.favro.net>

EXPERTISE/SKILLS

Programming Languages:

C++ & C, Lua, web [HTML/CSS/Javascript], Java, SQL, Objective-C, assembly (multiple architectures, 8-32 bit), shell scripting. I have worked in many languages, including Pascal, Fortran, Lisp, amongst others.

Frameworks/APIs:

Qt, JFC/Swing, NeXTStep/OpenStep AppKit (predecessor of Mac OS/X Cocoa), others.

Web development (front-end): some familiarity with *bootstrap* and *jquery*.

Designed and coded a cross-platform full-screen TUI package, including fields, menus, pick-lists, context-sensitive hypertext help, etc.

Many others, e.g. OpenMP, wrote a thin C++ wrapper around `libnetfilter_queue`, etc.

DBMS

- Developed using PostgreSQL, Sqlite, Sybase/MS-SQLserver, MySQL, and others, including custom-built database backends.
- Connectivity frameworks: Self-built rdbms/sql access abstraction in Lua as a layer on top of LuaSQL. JDBC, QSql, other custom-build adapter frameworks.
- Designed and coded systems using custom-built database access and management modules; implemented low-level storage and search techniques over both message-passing and shared-memory transports; client-server, peer-to-peer, and single-process.

Development & build tools:

Experienced in many VCS, build, development, and documentation tools, e.g. git, make, emacs, Java ecosystem, many others, including some custom-built.

Other expertise & experience:

NeXTStep, Microsoft Windows, MS-DOS, other mainframe and microcomputer OSs.

Very familiar with Unix, GNU/Linux, & POSIX system-call programming.

Linux and Unix (SysV & BSD) system administration, network administration, system security, sniffing & protocol analysis, cryptography, analog electronics design & prototyping, project management, team building and coordination. Design of various custom networking protocols such as single-roundtrip RPCs over UDP, etc.

System Administration:

Experience administering Linux servers (Debian & Redhat derived), Web Server, EMail server [MTA (Postfix), imap (Dovecot), custom-built highly configurable MDA], PostgreSQL, Asterisk (developed custom table-based call routing system), OpenVPN, DHCP server, DNS/BIND, Linux routing, most standard Unix/Linux facilities, filesystem/volume management (dm/LVM2/LUKS, raid/mdadm, etc.). Extensive shell scripting and system-administration via scripting in Lua using custom-built APIs.

EXPERIENCE

Recent projects:

For the last 10 years, as “side projects” I have been developing software (writing code nearly every day), primarily in Lua (i.e. my own forked dialect) as well as C++, for various personal projects. As Lua is not particularly well suited for general-purpose programming, I have built up a set of support module dependencies. The modules and applications are too numerous to list here, so only a few are enumerated.

- **EMail:** A complete stack to process email messages, including parsing messages to an internal object representation, API to access or modify all headers and parts of body, API to compose messages, serialization from internal to external representation, reading from and delivery to several formats (e.g. maildir, mbox), and complete MDA application with scriptable configuration.
- **Web Content Access:** An application to retrieve dynamic content from hundreds of sources, convert to a common format (extended markdown) via “adapters”, archive in a database, permit user to query, view, hierarchically categorize (tag), and attach notes via both web and command-line interfaces. Components include a library to assist in web-scraping.
- **Subprocess module:** a library to spawn subprocesses from Lua programs [the default Lua library has extremely limited ability]. Written in C but providing a Lua-native interface, allows launching processes [via `fork()` & `exec()` but using a pipe to monitor the child process’s `exec()` to report errors to the parent], read/write of subprocess input/output via pipes (including “bidirectional pipes”), chaining multiple child processes as a shell-like pipeline but without requiring any shell and using native Lua syntax to describe the processes to be created. Uses many Linux-specific optimizations [e.g. `splice()`, `sendfile()`, `pipe2()`] but falls back to Posix-compliant system calls when compiled for non-Linux kernels.
- **Web Server:** A nearly complete implementation of server-side http/1.1 [RFCs 2616 et al], including digest auth [RFC 7616].
- Many utility modules and applications, a few examples: user-options processing [CLI and config-file], error processing, “soft require” module lazy-loader, string processing, utf-8 string validator, text wrap for columnar display, filesystem interface, telecom interfaces, serialization/deserialization of objects, parsers for common file-formats such as ICS (vcalendar), networking utilities, a raspberry-pi based home-alarm system including a gpio abstraction layer in Lua, television program listing [retrieval, storage, query, display (CLI and web)], a billing system for contractor time, a bot to crawl Wikipedia pages, and bindings to third-party libraries such as `libbrotli`.
- **Open Source Contributions:** I have also made a few small contributions to open-source projects, e.g. Linux RAID.

2007 - 2012

Contract consultant, Center for Health Services Research (*now named Center for Individualized and Genomic Medicine Research*) at Henry Ford Health System, Detroit, MI.

- Developed an enhancement to a widely-used GPL human-genetics research modeling software to extend the model from individuals to pedigrees.
- Created lexical analyzer and parser for pedigree files and various other input files.
- Created various graph representation, traversal, manipulation structures and algorithms.
- The low-level state-transition modeling uses a Hidden Markov Model to evaluate likelihood of a given set of observed genotyped inputs and modeled parameters.

- The higher-level statistical modeling uses the Metropolis-Hastings algorithm to refine the modeled parameters to better fit the data via a random walk sampling.
- Extremely computationally intensive; I implemented concurrent shared-memory execution using OpenMP.
- Developed epidemiological research system:
 - Utilized by multiple research projects.
 - Implemented using C++, Qt, Lua, and PostgreSQL.
 - Platform-independent GUI front-end, runs on Linux, Windows, or OS X.
 - Participant tracking (CMS); participant surveys (flexible template-based data-entry forms with document-based results) integrated database with medical record data and specimen tracking.
 - Data-entry forms via custom-built runtime form-description module using sexp-based (Lisp) DSL format, custom parser, evaluator, and GUI-generator, with dynamically-built Qt presentation.
- Configure, install, and maintain computation, storage, and database servers for genomic and biostatistic research.
- See publications.

2005 - 2006

Contract consultant, CareTech Solutions, Troy, MI.

- Migration of 15-year-old legacy C++ application to modern compiler [gcc-4] and operating system [Linux, Solaris 10]
- System configuration of Solaris 10 on Sparc, network integration.
- Personally ported, tested, debugged 400,000 lines of C++ code.
- Developed plan for parallel testing and implementation.

2004/06 - 2004/10

Contract consultant, CareTech Solutions, Troy, MI.

- Contributed to the design and coded portions of a hospital information system in Java (JFC/Swing front-end, J2EE+WebLogic middle-tier, Sybase backend).

2004/05

Contract consultant, CareTech Solutions, Troy, MI.

- Developed complete requirements document for a middle-tier business-function orders management system, to be implemented in J2EE [short engagement].

2003/11 - 2004/01

Contract consultant, Henry Ford Health System, Detroit, MI.

- Provided analysis and consulting regarding the development and purchase of an order-entry, clinical data analysis, and clinical workflow system.
- Designed and coded a workflow engine in Java, accessible via SOAP web-services, utilizing JWSDP/JAX-RPC.

2002 - 2003

Contract consultant, Providence Hospital, Southfield, MI.

- Provided analysis, design, and coding of a prototype in-house developed decision support system, utilizing Borland C++ Builder™ under the Microsoft™ Windows® operating system.
- Designed and coded a prototype template-based documentation system in Java using JFC/Swing components.

2001

Contract consultant, Henry Ford Health System, Detroit, MI.

- Provided analysis and support for the purchase and implementation of a medical-records document-imaging system.

1989 - 1999

Director of Unix Systems Development, Mt. Clemens General Hospital, Mt. Clemens, MI.

- Director of the Department of Information Systems Development. Supervision of 8-10 employees including management employees. Built the department from the ground up; hired all employees.
- Designed, coded, and implemented the Patient Care System (PCS), a comprehensive, integrated enterprise-wide Hospital Information System application comprising order management, enterprise and operating-room scheduling, nursing care-planning and documentation including intra-operative documentation, template-based physician's documentation, clinical data repository, communications (GUI email MUA and instant messenger), electronic patient medical record, and many other modules, deployed onto over 250 workstations at 50 remote sites with over 1,500 users. Front-end in NeXTStep (objective-C using C++ libraries), back-end in C++ and Sybase RDBMS. Developed many utility-level modules including automatic software distribution, automatic centralized error logging, remote daemon control, etc. Believed to be the first enterprise-level health care client-server application, this system has been in operation for 10+ years and has received multiple international recognitions for its superior architecture, cost-effectiveness, and innovative adaptation of technology.
- Design and project manager of major software development projects, including: operating room system, patient care system (described above), multi-entity corporate general ledger system, fixed assets tracking system, cost accounting and management reporting systems.
- All development under the Unix operating system, on both NeXT workstations and iAPX386/486/Pentium machines, using C, C++, Objective-C, and other languages.
- Installed institution-wide 10base2 and 10base5 TCP/IP network, and custom developed many network management utilities.

1988 - 1989

Director of Financial Information Systems, (hired as Senior Financial Analyst) St. John Hospital, Detroit, MI.

- Promoted to Director of FIS, 2/89. Supervised 1 employee.
- Designed and coded financial and statistical budgeting system, A/P system, on-line payroll reporting system in C under SCO Unix and MS-DOS, and in Easytrieve under DOS/VSE. Wrote custom database server for budget application.
- Wrote terminal emulator under MS-DOS including interrupt-level RS-232 communications.

1986/06 - 1986/09, 1984/06 - 1984/09

Contract programmer, St. John Hospital, Detroit, MI

- Design and code financial reporting system and small payroll system in Pascal & Condor 4GL/RDBMS under MS-DOS, Easytrieve under DOS/VSE.

1985/06 - 1985/12

Programmer, Analysis & Technology, Middletown, RI.

- Coded modules of naval anti-submarine warfare software in FORTRAN-77 under HP-UX.
- Carrier-based system to analyze sonar data from multiple sources, predict optimal sonobuoy placement, run simulations of submarine movement, etc.
- US DoD security clearance.

1982 - 1983

Contract Programmer, Henry Ford Hospital, Detroit, MI.

- Designed and coded financial & database systems using FORTRAN, Easytrieve, and Focus under OS/MVS.

1982/06 - 1982/09

Student Intern, Wayne State University Physics Department

- Worked in imaging lab, coded in FORTRAN, calculate thermal responses of materials, produce 3-dimensional plots from results.

AWARDS

- Won by the PCS application developed at MCGH:
 - 1st place: 1994 DB/EXPO Intel RealWare Award for Best Enterprise Client/Server Application.
 - 1st place: 1994 Computer World Magazine and Object Management Group Best Use of Object Technology within an Enterprise or Large System.
 - 2nd place: 1998 Modern Healthcare First Annual Innovation in Healthcare Information Technology, Best Return-On-Investment (ROI).
- U.S. National Merit Scholarship, 1983
- 1st place: 1982 Michigan Math Competition for High-School Students
- 3rd place: 1981 Competition in Mathematics, Lawrence Technological University

EDUCATION

B.A. in Mathematics, Brown University

Providence, RI., May, 1987. Computer science topics include data structures, analysis of algorithms, compiler design, artificial intelligence, database design, etc. Mathematical topics include number theory, linear algebra, functional analysis, differential geometry, complex analysis, partial differential equations, topology, abstract algebra, probability & statistics.

SELECTED PUBLICATIONS

- Extending Admixture Mapping to Nuclear Pedigrees: Application to Sarcoidosis, *Genetic Epidemiology*, Jan 2013
- Genetic variation in B cell-activating factor of the TNF family (BAFF) and asthma exacerbations among African American subjects, *Journal of Allergy and Clinical Immunology*, Jun 2012
- Quantifying the proportion of severe asthma exacerbations attributable to inhaled corticosteroid nonadherence, *Journal of Allergy and Clinical Immunology*, Dec 2011
- Persistence of Worsening Renal Function during Acute Heart Failure Hospitalization, *Journal of Cardiac Failure*, Aug 2009
- Differences in allergic sensitization by self-reported race and genetic ancestry, *The Journal of Allergy and Clinical Immunology*, Oct 2008

MISCELLANEOUS:

Spoken/written Languages: native English speaker; nearly fluent in French; proficient in Vietnamese.

References & example code available upon request.

commit:7b50aa3d76