Daniel Sommers

1407 Woodwind Court
Corbin, KY 40701
dan@tombstonezero.net
+1 606 523 5592

Career Objective	A position of technical leadership in full life cycle software development, focusing on specification, design, and implementation of complex systems
Operating Systems	Unix/Linux, Mac OS, Windows, other proprietary real-time and non-real-time operating systems
Programming Languages	C, C++, Unix Shells, Python, Java, MC680x0 Family Assembler, 80x86 Family Assembler, FORTRAN, PASCAL, lex, yacc, sed, awk, others
Patent	Co-Author, U.S. Patent 5,291,488, March 1, 1994, Communication System Configuration Data Management. Abstract: System configuration data required for use by various interfaces can be obtained either directly via a data link that couples the interfaces to other systems, or by a down link from a system data manager configured as part of the interfaces' system.
Publication	Co-Author, Standing Stones, Electronic Arts, 1983, a fantasy role playing game for the Apple II and Commodore 64 computers.

Education

University of the Cumberlands, Williamsburg, Kentucky — *August 2002 through May 2006*

Bachelor of Science with majors in Mathematics and Physics.

Member in good standing of Kappa Mu Epsilon and Sigma Pi Sigma.

Experience

Science Applications International Corporation, NIMS SC — *May 2006 to Present*

Brought development of Dashboard, a web application summarizing national emergency preparedness, in house. Led two releases successfully through requirements, design, testing, and deployment.

Led development of Incident Resource Inventory System (IRIS), a distributed inventory system for emergency management resoures. IRIS is presently distributed by FEMA.

Provided the technical point of view as this organization developed its internal software development processes.

Windows, Linux, Subversion, Java, IIS, ASP, Visual Basic, SQL Server, XML, EDXL.

Motorola, High Availability Platform* — *September 2000 through April 2001*

Led team to extract software requirements from marketing requirements, system engineering requirements, and ISO/CCITT/ITU recommendations. Produced textual and UML Software Requirements Specification.

Led team to develop UML-based Software Interface Design Specification using Rational Rose.

Participated in various configuration management process improvement efforts to help the department use Rational ClearCase and ClearQuest.

Provided technical leadership for the High Availability Applications team, including Event Management, Alarms Management, Statistics Management, and Upgrade Management, through the first major release.

Established future design and implementation directions for the Alarms Management application.

Uponus Techonologies *December 1999 through August 2000*

Took over previously out-sourced development of Mac OS and Windows compression and encryption software. Brought proofs-of-concept to market.

Developed proprietary implementations of public and proprietary compression and encryption algorithms for comparison purposes using Metrowerks CodeWarrior, C, C++, Python.

Motorola, Iridium ECS* *January 1997 through November 1999*

Specified and designed advanced call processing features (e.g. call waiting, three-way calling, etc.) in text and in SDL, produced prototype, implemented and tested portions thereof.

Led team to retrofit proprietary multi-tasking MC68040 operating system into another application fork.

Specified and designed periodic polling and fault management of unintelligent fiber-optic daughter boards after redesigning and re-implementing entire RS-232 communications subsystem.

Developed many internal development tools, process improvements, high level design document standards, and C language coding standards.

Unix (Solaris and SunOS), Software Through Pictures, Object Geode, SDL, C, MC680x0 Family Assembler, Python development environment; custom, lan-based multi-processing MC680x0 hardware target.

This organization was assessed SEI Level 5.

Motorola, Iridium OMC-R* *January 1996 through December 1996*

Specified, designed, implemented, and tested SNMP manager and agent for Iridium OMC-R (Operations and Maintenance Center, Radio). The OMC-R was an SNMP agent to the Iridium OMC-G, and an SNMP manager to the Iridium ECS.

Developed many build utilities and other SCM tools.

Motorola, Network Services* *January 1990 through December 1995*

Led SNAP (SMR Network Administrator's Package) development team.

Gathered requirements, specified, designed, developed, tested, and deployed SNAP, a Unix-based Operations and Maintenance system for SMR networks. SNAP's primary functions were to maintain a local database of authorized SMR users, and to keep up to 1000 remote SMR controllers in sync with that database. Unix (Apollo Aegis) and C development environment, separate Unix (System V) target. In late 1999, SNAP's target platform was declared non-Y2K compliant; SNAP had been part of daily operations with neither maintenance nor support since mid-1994.

Gathered requirements and specified SNAP's successor, eventually assisted third party with design, development, and deployment.

Developed many in-house Unix-based utilities for various uses throughout the Network Services organization.

Motorola, Control Centers* *January 1989 through December 1989*

Ported existing proprietary MC68000 operating system to newly designed MC68020 hardware. Worked with hardware engineers to debug hardware and low-level software.

Specified, designed, implemented, and tested distributed database functionality (see patent, above).

Specified, designed, implemented, and tested new communications drivers, stacks, and protocols for expanded packets necessary in a networked environment.

Unix (Apollo Aegis), DSEE, C, Pascal, MC680x0 Assembler, MC6803 Assembler development environment; proprietary MC680x0 and MC6803 targets.

*American College of Surgeons** *October 1988 through December 1988*

Designed and implemented software for grading college members' self-assessments in FORTRAN on an IBM 370.

Designed, implemented, and tested software for extracting oncological information from a proprietary database for subsequent use in a DB/2 application. MS-DOS and C development environment, MS-DOS target.

Motorola, Trunking Systems* *July 1988 through September 1988*

Specified, designed, implemented, and tested CCROSS, a system for synchronizing multiple radio towers in a simulcast network. Developed our own protocols for use across half-duplex microwave and RS-485 networks. Unix (Apollo Aegis), C, and MC68HC11 Assembler development environment; IBM-PC and proprietary MC68HC11 targets.

CompuQuote *April 1985 through June 1988*

Designed, implemented, tested, and deployed real-time and off-line stock, option, and future analysis software, including real-time pricing, historical graphs, and risk analysis and management. Worked with traders and trading staff to develop proprietary theoretical value models, custom graph types, and other analysis tools.

Z80-based network, client-server architecture, distributed data base, everything from low-level FIFO, RS-232, Weitek 4167 and Intel 80387 FPUs, and video drivers and libraries to multitasking operating system and applications. CP/M development environment; CP/M, MS-DOS, and proprietary and off-the-shelf Z80 family hardware and software targets; PL/M, C, Z80, and 80x86 Family Assembler languages.

B.I.E.S. Systems *July 1982 through April 1985*
Byte Shop *October 1978 through June 1982*

Built, sold, repaired, and wrote software for a variety of CP/M-based computers, Apple IIs, IBM-PCs, and other microcomputers of the time.

Taught myself BASIC, Pascal, FORTRAN, C, and 6502, 8080/8085/Z80, MC680x0, and 80x86 family assembly languages.

References Available upon request.

*These work experiences were as a contract employee of Interactive Business Systems, Oak Brook, Illinois.