

R É S U M É

MARK J GLENN

SOFTWARE ARCHITECT · CONSULTANT

PROFILE

- Career Software Architect and Consultant who enjoys writing small, fast, elegant software. Prefers long-term projects, building from the ground up, from inception through product delivery. Works independently or with your team.
- Creator of the real-time messaging application sked.skccgroup.com.
- Creator of OOSMOS, the **O**bject-**O**riented **S**tate **M**achine **O**perating **S**ystem. www.oosmos.com.
- Former member of the Software Defined Radio Forum (SDR Forum) representing Motorola.
- Former member of X3J16, the ANSI C++ standardization committee.
- Co-inventor on the patent for Covidien's dual-head power injection system supporting simultaneous injections. (<https://www.google.com/patents/US20130245439>)

SKILLS

RECENT	OOSMOS, TypeScript, Docker, Node.js, WebSockets, Python, SQLite, networking, state machines, and Git.
EXPERT IN	C, C++, IBM's Rhapsody, hierarchical state machines, large-scale software architecture, object-oriented design, real-time embedded systems, performance optimization, Perl, PHP, real-time CORBA internals, and compiler design.
PROFICIENT IN	Java, UML, VBA, HTML, XML, CSS, Assembly, Win32, Unix/Linux, MS-DOS, pSOS, VxWorks, Integrity, MQX, Liberty OS, ClearCase, databases (SQLite, MySQL, Access), SQL, PEG, x86, 68xxx, DSPs, Software Defined Radios (SDR), PCI bus, TCP/IP, and sockets.
EXPERIENCE WITH	C#.

WORK EXPERIENCE

OOSMOS, LLC

Phoenix, AZ

March 2013 to Present
President, Chief Scientist

- Creator of OOSMOS, the Object-Oriented State Machine Operating System. www.oosmos.com
 - Designed and implemented the complete operating system.
 - Implemented the statechart-to-C code generator in TypeScript.
 - Implemented the OOSMOS website using a custom object-oriented PHP framework. The website is completely responsive, meaning the same set of code supports both desktop and mobile devices.

Stress Engineering Services, Inc.

Mason, OH

March 2017 to April 2018

Consultant

- Using the OOSMOS operating system targeting an STM32 ARM processor, designed and implemented the software for a wound suction medical device. The device obtained FDA clearance on the first pass.

Reliance Medical Products

Mason, OH

August 2012 to March 2013

Consultant

- Reimplementation of two medical devices: an ophthalmic exam chair and an instrument delivery system. Written in object-oriented C on the PIC32 platform using the OOSMOS operating system.

GE Healthcare

Milwaukee, WI

May 2010 to July 2012

Consultant

- Working with GE Healthcare's Principal Engineer, designed and implemented a Rhapsody-based distributed object framework to support implementation of GE Healthcare's next generation CT scanner.
- Using Rhapsody's Java API, designed and implemented an automated mechanism to generate communication proxies from interface definitions specified graphically in a Rhapsody model.
- Acted as principal Rhapsody mentor to quickly get the team using Rhapsody's advanced features effectively.
- Provided initial design for the Axial Rotation Subsystem, the subsystem that drives the large rotating element of the CT scanner gantry.

Covidien

Cincinnati, OH

January 2006 to November 2009

Consultant

- Using Rhapsody, OXF, C++, UML, and PEG, redesigned and reimplemented a dual-syringe injection medical device.
- On an existing power injector, rearchitected all motor control elements and directed performance improvements that reduced CPU usage from over 100% to less than 25%.
- Designed and implemented a motor control PID algorithm to assure safe pressure levels of fluid within syringe during injections.

Northrop Grumman

San Diego, CA

April 2005 to December 2005

Consultant

- Established an end-to-end software development process using Perl on top of ClearCase, including automated builds of Rhapsody models.
- Developed and presented advanced CORBA C++ training to JSFCNI developers.
- Regular presenter of software engineering best practices to JSFCNI developers.

Tyco Healthcare

Cincinnati, OH

February 2005 to April 2005

Consultant

- Tuned an injection device's motor control PID algorithms, written in C++, to detect and react to anomalous pressure in real time.

Northrop Grumman/PrismTech, Ltd

San Diego, CA/Newcastle, England, UK

January 2004 to November 2004

Consultant

- Worked with PrismTech and Northrop Grumman engineers to identify and mitigate high-risk CORBA performance and transport-related issues with existing C and C++ CORBA ORB technology.

- Ported the C-based CORBA ORB I wrote while at Motorola to a proprietary processor selected by Northrop Grumman's JSFCNI program and implemented significant performance optimizations. Metrics are proprietary.
 - Prototyped a C++ CORBA ORB, targeting Windows and a proprietary processor.
-

Glenn Software Consulting, Inc.

Phoenix, AZ

Mar 2003 to Dec 2003

E-commerce Web Developer

- Using PHP, JavaScript, AJAX, and MySQL, developed e-commerce websites including associated content management systems.
-

Northrop Grumman Electro-Optical Systems

Tempe, AZ

Oct 2002 to Feb 2003

Consultant

- Using C++ and Visual Basic, upgraded a test station that used fiber-optic communication for a proprietary military high-altitude thermal camera application.
-

Accenture

Minneapolis, MN

July 1999 to May 2002

Consultant

- Designed and implemented a message exchange system in C++, XML, and CORBA for Accenture's Launch-Now startup venture.
 - Designed and implemented a complete work-packet and life cycle configuration management system in Perl on top of ClearCase.
 - Designed and implemented a complete general-purpose database management system in Perl on top of Oracle.
-

Motorola SSPD, Wireless Information Transfer System (WITS)

Scottsdale, AZ

January 1998 to July 1999

Consultant

- Designed and implemented a real-time CORBA ORB for C, the foundation of Motorola's WITS software-defined radio distributed object architecture. This 'C' ORB was 3 times faster than Washington University's 'TAO' ORB. It ran on NT and VxWorks, as well as on a SHARC DSP (a first in the industry).
 - Designed and implemented the CORBA IDL-to-C compiler, written in Java, to complement the WITS ORB.
 - Co-developed the WITS object-oriented radio framework in C. Worked with Motorola's Chief Scientist to introduce this framework to industry for standardization by the Software Defined Radio (SDR) Forum. This framework was the earliest form of today's SCA.
 - Developed the WITS radio simulation platform on Windows, able to simulate the operation of the entire radio. This platform was the principal WITS testing platform.
 - Designed and implemented a real-time, write-only, PCI-based shared memory OSI transport layer in C.
 - Technical lead of the WITS software infrastructure, responsible for the CORBA ORB, Radio Control, RF, and SNMP components, all written in C.
 - Developed and evangelized the WITS software architecture strategy: Distributed object technology using CORBA.
-

Motorola SSTG, SPEAKeasy II

Scottsdale, AZ

November 1995 to January 1998

Consultant

- Contributing system designer and implementer of the Motorola SPEAKeasy multi-band multi-mode reprogrammable radio successfully demonstrated at military training exercises at Task Force XXI in March 1997.
- Software lead of the System Control and INFOSEC product teams.

Platinum Technology

San Francisco, CA

May 1995 to Nov 1995

Consultant

- Implemented major modifications to the Microsoft MFC object-oriented class framework to lift severe 32K object size limitations. The modified MFC was used to implement Platinum's successful SQL-Coder database management application.
- Ported the C++ Windows-based SQL-Coder application to UNIX.

Andersen Consulting

Minneapolis, MN

Apr 1992 to May 1995

Consulting Technical Project Leader

- Lead designer and co-implementer of Andersen Consulting's Universal Construction Toolset (UCT), Andersen's client/server middleware that allowed rapid development of GUI, batch, and server applications for Andersen clients on a wide range of platforms.
- Designed and co-implemented UCT's object-oriented GUI in C++.
- Implemented a complete, generalized data entry, validation, translation, and editing library for date/time, floating-point, and masked data entry in C++.
- Designed and implemented a generalized, reentrant, automated testing system API in C++. Using this testing interface, programmers could build into the system self-verifying tests that executed at startup, before main() was called.
- Designed and implemented a complete, generalized, object-oriented date and time library in C++. Features include symmetric format strings for parsing input and formatting output, daylight saving time support, time zone conversions, extensive date/time arithmetic, entire Gregorian calendar support, and date/time validation.
- Designed and implemented a suite of server self-tests in C++. Traditionally, server tests involve complicated setup of at least one client and one server process, possibly on more than one machine. This server test was a single, self-contained, multi-threaded client and server executable that "talked to itself," reducing setup and maintenance time.
- Designed an integrated exception handling mechanism. Not all compilation systems that UCT used supported the C++ exception handling mechanism.
- Developed and wrote the project's C++ coding standards.
- Designed and implemented a multi-database API in C++. Supported two implementations: Sybase and Oracle.
- Designed and implemented a complete Microsoft Excel-compatible data formatting library in C++.
- Designed and implemented a complete, generalized "canonicalization" library in C++ to handle the necessary byte-swapping and type promotion to allow data transfer between machines of different architectures in a heterogeneous environment.
- Designed and implemented a generalized memory debugger for C++ by overloading the 'new' and 'delete' operators. The system could detect memory leaks, corruption, reads of uninitialized or freed memory, and multiple frees of the same memory area.
- Designed and implemented the primary mechanism that made UCT so portable: using C++'s ability to support multiple implementations under a single interface, rather than the traditional use of #ifdefs.
- Designed and implemented a portable startup mechanism that allowed C++ objects to be constructed and destructed in a predictable order in the "shared library" environments of Windows and UNIX.
- Co-designed a portable socket communication interface in C++ that cooperated with UCT's multi-platform multi-threading library. Supported Winsock for Windows and Unix sockets for both SVR4 and Berkeley.
- Ported UCT to Windows 3.1 and AT&T UNIX SVR4 platforms. Assisted others with ports to all other platforms.
- Taught classes on C++ and object-oriented design. This project was Andersen Minneapolis' first exposure to C++ and object-oriented software development.

Sun Microsystems

Phoenix, AZ

Sep 1991 to Apr 1992

Consultant

- Joint venture between Sun Microsystems and Andersen Consulting to develop Andersen's next generation of client/server C++ middleware, now called UCT (Universal Construction Toolset).
- Assisted developers in transition from C to C++, introducing them to object-oriented design and development.
- Designed and implemented a program generator in C++ for UCT that read a design database and produced ANSI C source code. Programmers then augmented this code with business logic.

Loral Defense Systems

Goodyear, AZ

May 1991 to Sep 1991

Consultant

- Integrated a landing system for an unmanned "drone" helicopter in C++.
- Implemented in C++ a multi-board device driver for the Motorola MVME332 Intelligent Communication Controller.

Glenn Consulting

Phoenix, AZ

Feb 1989 to Apr 1991

Software Engineer

- Designed and implemented, in ANSI C, an ANSI C compiler for the 680xx-based Amiga personal computer.
- Implemented a complete ANSI C-compliant preprocessor in ANSI C. It was the only one of eight competing compilers that executed the examples in the standard correctly.
- Implemented a complete ANSI C-compliant run-time library, including memory management, standard I/O facilities, floating-point math routines, and date/time manipulation.
- Implemented an ANSI C test suite to test coverage of features, global name space pollution of the run-time library, and "edge" conditions that many compilers fail.

Loral Defense Systems

Goodyear, AZ

May 1988 to Feb 1989

Consultant

- Designed and implemented hardware interface libraries for a radar system in C++.
- Taught introductory C++ classes to project members.

Honeywell Bull, Inc.

Phoenix, AZ

Apr 1988 to May 1988

Consultant

- Designed and implemented a high-level structured, custom script language in C++ for Microsoft Windows and MS-DOS.

DDC International, Inc.

Phoenix, AZ

Oct 1987 to Apr 1988

Senior Software Engineer

- Co-designed the DDC-I VAX-to-80x86 Ada run-time system for real-time embedded system applications. Primary objectives were minimal size of required code and data space as well as fast execution of memory management and the Ada tasking model. The result was the smallest and fastest Ada run-time environment in the industry.
- Acted as technical liaison between the Phoenix and Copenhagen offices, spending 15% of my time in Denmark.

- Lead software designer and implementer of an IBM PC compatible high resolution graphics workstation.
- Implemented a complete character-based windowing system and user interface in C.
- Implemented MTX, a portable real-time multitasking executive in C. MTX ran under UNIX and MS-DOS, and as a stand-alone operating system suitable for embedded systems.
- Implemented a chip-level device driver on an IBM PC for an ITT-proprietary coax PC board in C. The driver used MTX to meet its real-time constraints.

- Implemented in C the XNS network protocol on an IBM 4341 running VM/CMS for international factory automation applications.
- Improved by over 600% the performance of an existing C multitasking system for use in a virtual machine running under VM/CMS.
- Implemented network performance analysis tools in C to verify optimal network configuration and performance.
- Developed a high-level interprocess message communications interface to allow C and Pascal programmers on VM/CMS to transfer data between virtual machines.

- Designed and implemented a UCSD Pascal-compatible compiler for the TI 9900 microprocessor-based 99/8 computer. Written in Pascal, it included global optimization using flow analysis, elimination of redundant range checking and subexpressions, and support for symbolic debugging.
- Supported the interpreter for GPL (Graphics Programming Language), the language used by most Texas Instruments Home Computer programs.

- Co-designed a multi-environment, globally optimizing COBOL compiler, capable of generating code for the Honeywell CP6, GCOS, and Multics operating systems. Implemented the code generator and code generation related parts of the front-end.

.....- - - - -.....- - - - -.....- - - - -.....

H O B B I E S

- Amateur radio operator. Extra class. Exclusively HF Morse code. Callsign K7MJG.
- Flight simulation. (X-Plane, RealSimGear G1000.)
- Private Pilot. (Not current.)

.....- - - - -.....- - - - -.....- - - - -.....

E D U C A T I O N

Ohio University, Athens, Ohio

BS Computer Science, Cum Laude, 1979