Paul S. Heidmann
5218 N. 69th Ave.
Glendale, AZ 85303
Cell: (602) 818-5385
paul@heidmann.com
http://heidmann.com

Education

- M.S., Mathematics, 1992, North Dakota State University, (GPA 3.94).
- B.S., Mathematics, 1990, North Dakota State University.
- B.S., Electrical Engineering (Computer Option), 1990, North Dakota State University.

Publications

- <u>November, 1993:</u> "A Statistical Model for Designers of Rate Monotonic Systems", Proceedings of the Second Annual Rate Monotonic User's Forum (hosted by the SEI).
- <u>September, 1997:</u> "Transient Analysis of Ocular Drug Delivery, Zero Volume Effect", Journal of Pharmaceutical Sciences (Coauthors: J. C. Keister, Paul Missel).

Experience Summary

Hardware
ARM, Intel, PowerPC, TI 6713, VME, MIL-STD-1553, RS232, RS422, Logic Analyzers, Motorola 68360

Software Languages
C++ (including C++11, boost), C, Java, Ada (95 and 83), Assembly

Operating Systems
DeOS, LynxOS, vxWorks, pSOS+, Linux, Solaris, Windows, FreeBSD

CASE Tools
Git, Subversion, Doxygen, Eclipse, Rational Rose, Rational Clearcase, Greenhills AdaMULTI, CVS

Experience

Geco

<u>Nov/2017 – Present</u>: Designed and implemented code generator to automate (completely) the generation of a FACE (Future Airborne Capability Environment) TSS (Transport Services Segment) from an arbitrary model. Designed and implemented DLEP (see RFC 8175) router. Maintained and significantly extended a Qt based GUI with interactive map, symbology, phonebook, and radio controls. Most code executed on both Intel and PowerPC.

SAIFE

<u>Jan/2016 – Nov/2017</u>: Software Engineer. Worked on a team that developed a VPN solution using SAIFE's cryptographic and communications SDK and SAIFE's continuum servers. Provided all IPv6 and Software Defined Perimeter functionalities. Increased VPN's throughput by a factor of ten. Also maintained and added to SAIFE's SDK. All code written in C++11 (including boost). Minor amounts of Java.

Equinox Payments

<u>Dec/2014 - Jan/2016</u>: Diagnostics Software Engineer. Designed and developed code to be used by technicians to test hardware as it is produced, for warranty returns, for compliance testing, and for validation engineering. Also refactored a large codebase of poorly written diagnostics code. The code developed/refactored goes from the device driver level (where the hardware tests are performed) up to the GUI level (technician interface). The greatest majority of the code written in C++11 (with the

lowest level code written in C). The GUI applications use Qt. The embedded code executes on an ARM platform running busybox Linux, while the GUI code executes on a Linux PC.

DDC-I

Jan/2014 – Dec/2014: Software Engineer. Designed and developed embedded, hard real time software primarily for avionics platforms. Implemented Ethernet drivers, maintained an embedded TCP/IP stack (LwIP), implemented board support packages, while using JTAG emulators. Much of the work was to DO-178B/C standards. Used x86, PowerPC, and ARM processors.

Equinox Payments

April/2013 - Jan/2014: See the description of my current position, with Equinox Payments, above.

iTRACS

September/2011 – April/2013 : Software Engineer. Designed and developed web services using C++11, axis2, SOAP, and the Apache stack. Designed and developed XML parsers using the Xerces library. Making custom modifications to the Apache stack (via Apache modules). The greatest majority of the code is developed in Visual Studio 2010, with minor amounts of code developed on Linux. Used the new C++11 language extensions heavily.

Wells-Fargo (independent contractor)

July/2011 - September/2011 : Software Engineer, Storevision Teller (the application used by bank tellers). Designed, developed, and maintained both GUI and backend code. Used Qt extensively. All code executing on Linux.

General Dynamics

May/2006 - May/2011 : Technical Lead, FCS Program. Design, develop, and debug sensor communications code. Created SQL and XML driven build process with auto-generated code. Used Xerces C++ library. Code written in C++, C, and SQL (triggers). Sensor code interfaces with SOSCOE and executes on TI 6713 DSP (O/S: TI BIOS), PowerPC (O/S: vxWorks), and Intel (O/S: Linux and LynxOS). Script code executes on Linux and on Windows.

Dec/2005 - Apr/2006 : Software Architect, Prophet Program. Participated in proposal process. Specified software products, worked with subcontracting companies, wrote SOWs, requirements, and specifications.

March/2005 - Nov/2005 : Technical Lead, FCS Program. Developed sensor communications software in Java and C. Interfaced with SOSCOE. Code executed on Linux and on a TI 6713 DSP.

Nov/2004 - March/2005 : Software Engineer, TSP Program. Developed sensor management software. Interfaced with SOSCOE, employed WSDL, XML. All code written in Java.

Aug/2004 – Nov/2004 : Software Engineer, CAC2S Program. Developed and integrated a recorder of network traffic. Made use of CORBA (the Tao orb) and ACE (Adaptive Common Environment). All code written in C++ using Microsoft's .NET IDE.

Jan/2004 – Aug/2004 : Software Task Lead, Land Warrior program. Overseeing three subcontracting companies, ensuring that they develop software in compliance with a CMM level three process. Providing technological and architectural oversight for all phases of software development for these three subcontracting companies. Specified and constructed a Linux and Solaris based isolated LAN.

May/2003 – Jan/2004 : Software Architect, Land Warrior program. Overseeing software development of three subcontracting companies. Software architecture definition and documentation with Rational Rose, including use case development, activity diagram contributions, sequence diagrams, class diagrams. Mentored subcontractors in CMM level three compliant processes (note that General Dynamics is a CMM level five organization).

Feb/2002 - May 2003 : Software Architect, MCS program. Overseeing 8-10 engineers in the development of software for a software definable cryptographic system. Defined and developed the S/W architecture. Developed Ethernet subsystem.

Xyterra (no longer in business)

Nov/2001 - Jan/2002 : Software Engineer developing customized routing protocols to be used to route IP traffic around failed compute nodes. Worked with Ethernet drivers. Integrated these and several other Linux modules into the Linux O/S on a PowerPC system. All code written in C.

Motorola (independent contractor)

Dec/2000 – Nov/2001 : Software Engineer, Lynx project. Developed and integrated ipSec and SP3 (Security protocol 3). Integrated these security protocols into IP, and developed custom IP multicast/broadcast solutions. Some BSP work with the VxWorks operating system. Code written in C/C++.

Honeywell (independent contractor)

Mar/2000 - Dec/2000 : Software Engineer, Radio Systems Group. Developed a specification for a proprietary serial bus. Did systems level requirements work. Designed, wrote, and integrated simulation software for the AV-900 Radio Controller (used in commercial aircraft), using the VAPS (Virtual Prototypes) graphical system and Microsoft Visual C++. Converted Intel 87C51 micro-controller assembly language code into C.

Boeing (independent contractor)

Jan/1999 - Mar/2000 : Software Engineer, Apache Training Systems Group. Designed and wrote software for the Flight Controls System Trainer (FCST). Performed hardware/software integration on the FCST, and delivered it to Boeing. Also designed, wrote, and integrated the following: Load maintenance panel (RS422), area weapons system interface, video switcher (RS232), and type 1B serial bus driver. All software was written in Ada and integrated on PowerPC VME boards running VxWorks (Tornado). CASE tools used: Rational Rose and Green Hills compilers and debuggers.

Ensco (on site at Motorola)

Jan/1998 - Dec/1998 : Software Engineer, NSM (Network Security Manager) program. Designed, wrote, and integrated the X.500 directory interface. All software written in C++.

August/1997 - December/1997 : Software Engineer, Land Warrior program. Designed, wrote, and integrated the system messaging system, the digital to analog converter drivers, and the X.21 communications drivers. Also integrated the TCP/IP stack. Software was written in C and C++ and integrated on 68360 based boards running pSOS+.

Computing Devices International

Jan/1997 - Aug/1997 : Software engineer and Technical Lead (overseeing 8 engineers), Boldstroke program. Implementing a CORBA (Common Object Request Broker Architecture) compliant ORB (Object Request Broker) on the following platforms: WindowsNT (using Microsoft Development Environment), Solaris, and a PowerPC based VME board (using Green Hills Tools). All code is in C++. Made heavy use of Rose.

Mar/1996 - Dec/1996 : Principle investigator, AIMS internal research and development program. The AIMS program successfully designed and developed a generic avionics Applications Program Interface (API). It also developed a digital map application (used to test the generic API). All target side code was written in Ada, all host side code (image processing) in C++. Object oriented methodologies and tools (Rational's Apex and Rose) where heavily employed. In this effort I acted as technical lead to four engineers and assumed many of the project management duties.

Dec/1993 - Mar/1996 : Software engineer, LAMPS program. Designed, wrote, and integrated the following CSCs: SCSI (driver/client/server), Interrupts, Timers (included multi-module clock synchronization), and Discretes (driver/client/server). I was also heavily involved with the following activities: Futurebus+, the run-time executive, and TM bus. All software was written in Ada. Software Through Pictures employed.

Sep/1993 - Dec/1993 : Worked on an internal research and development effort (conceived by myself) to combine statistical analysis with Rate Monotonic Analysis to allow risk analysis in the early stages

of a program. Authored a paper that was published in the proceedings of the Second Annual Rate Monotonic User's Forum (hosted by the SEI).

<u>Dec/1992 - Sep/1993</u> : Test engineer for the F-14 program. Produced drivers for a 1553 board and extended memory (on a PC). Code allowed for real-time configuration of the test hardware and software based on commands received over the 1553 bus (written in C++).

<u>September/1992 - December/1992</u> : Worked on an internal research and development program studying Rate Monotonic Analysis (RMA). Authored a paper surveying current theory and practices. Produced tools used to determine schedulability of code from profile outputs (code written in Ada).

Clearance

Active Secret, Former TS/SCI