Raymond Lambert

Senior/Principal Software Engineer

530 Grove Street, Framingham, MA 01701
☎ *774-279-2999*
✉ *raylambert@interthingy.net*
🖱 *http://interthingy.net/ray/resume/*

Skills Summary

Languages • C++ • C • Python • Bash • HTML • CSS • PHP • JavaScript • jQuery • Java • SQL • assembler • *many others...*

Systems & Tools • Linux • Unix • POSIX • Apache • Tomcat • Git • Mercurial • wxWidgets • Qt • GTK+ • Windows • Cygwin • Embedded Systems • *many others...*

Technical Skills • Open Source tools and development methodologies • GUI design and development • Object-oriented design • Strong technical documentation skills • SCM • Build automation • Tool-smithing

Personal Skills • Excellent written and interpersonal communications • Rapidly assimilates new technologies and environments • Works well in groups or individually • Well organized and self-motivated • Creative problem solver

Professional Experience

Aug 2002 – present **Volpe National Transportation Systems Center (USDOT)** – Cambridge, MA
🖱 https://www.volpe.dot.gov

Job Title Information Systems Specialist

Roles Principal Software Engineer; Lead Developer; System Architect

Technologies Used • C/C++ • Python • Bash • Git • Ruby • Perl • Java • Tomcat • HTML/CSS/JavaScript/jQuery • PHP • Linux/Solaris/AIX • Oracle • MySQL • Windows • TCP/IP networking

Details Developed new and maintained existing software for various Federal Aviation Administration (FAA) programs:

- From January 2013 to the present, served as a member of the ADS-B (Automatic Dependent Surveillance-Broadcast) SAPT (Service Availability Prediction Tool) team that is developing and maintaining two web sites that provide official FAA information services to pilots and airlines about the availability of GPS-based navigation capabilities along aircraft routes-of-flight. The web sites use HTML, CSS, JavaScript, jQuery and PHP on the front-end and Java on the back-end, including Tomcat-based Java Web Services and standalone back-end servers written in Java, and an Oracle database:
  - 🖱 https://sapt.faa.gov/ – Predicts the availability of GPS service and the performance of ADS-B-based navigation for aircraft along their routes-of-flight. The web site offers both an interactive HTML-based web form as well as an XML-based web service designed for automated use by flight planning systems. New FAA rules (which are currently out for public review and planned to go into effect in 2020) will require all flights that do not meet minimum hardware standards to use this web site pre-flight to determine whether they will meet the required navigational performance.
  - 🖱 http://www.raimprediction.net/ – Predicts the availability of GPS RAIM (Receiver Autonomous Integrity Monitoring) for aircraft using certain classes of navigation hardware. Proper use of this tool satisfies an FAA requirement for these aircraft when operating on certain designated routes and is explicitly recommended by the FAA for this purpose. The web site offers both an interactive HTML-based web form as well as an XML-based web service designed for automated use by flight planning systems.

    The RAIM Prediction web site was previously developed by another team and its maintenance and further development was turned over to the SAPT team in early 2013.

    In 2015, raimprediction.net was discontinued and its functionality was integrated into sapt.faa.gov.
- From November 2012 to January 2013, developed a data collection server to capture data from weather instruments via a 32-port serial card, a *counter card* and a frame-grabber device. The software was developed in C++ in a Linux environment and deployed on a Windows 7 system under Cygwin. The software is part of an experimental weather monitoring program for aviation at Otis Air Force Base and serves as the back-end for an experimental web site that provides a *dashboard* view of the data.

- From March 2009 to November 2012, served as Lead Developer and System Architect on a project to develop a Monte Carlo-style, fast-time aircraft/vehicle simulation tool. Developed the software architecture and led several developers in implementing the software suite (C++ and Python).

  The tool had a focus on ground-based operations (with some airborne operations) and was designed specifically to analyze a new concept for reducing taxi delays on the airfield called Collaborative Departure Queue Management (CDQM). The basis of CDQM is to predict taxiway congestion and to mitigate it by holding some aircraft in the gates for some period of time (which also reduces the amount of fuel wasted while idling on a taxiway).

  The simulator generated synthetic tracks (aircraft), based on a defined scenario, that could autonomously navigate the airport surface and negotiate traffic situations. It was able to integrate with the CDQM server while simultaneously acting as a surveillance data source (i.e. track data) and a consumer of CDQM advisory data, which it used to control the release of departure aircraft from gates. Performance analysis of CDQM was conducted by statistically comparing the airfield performance for the given simulation scenario both with and without input from CDQM.

- From September 2006 to March 2009, served as Lead Developer and System Architect on a project to develop a Monte Carlo-style, fast-time aircraft simulation tool focused primarily on airborne flight in and around airport terminal areas. Developed the software architecture and led several developers in implementing the software suite (primarily in C++).

  The tool was intended to replace a sponsor's existing tool and therefore was required to duplicate certain functionality as well as implementing new functionality. The tool is used by the FAA sponsor to study proposed changes at the Nation's airports and to make formal determinations regarding the safety of the changes as well as formal recommendations for or against the changes. The types of changes that are studied include physical changes (e.g. new runways), configuration changes (e.g. traffic patterns and uses of runways and taxiways) as well as proposed rule changes related to airport operations. All such studies are focused on safety and, in particular, on the mitigation of unexpected circumstances such as unplanned aircraft movements.

- Prior to September 2006 served as Principal Software Engineer on several smaller projects:

  - Maintained a prototype airport surface surveillance system for the FAAs SafeFlight21 program. The system fused data from multiple sources (including ASDE-X multilateration, secondary radar, flight plan database, and aircraft transponders) to generate a live, real-time "picture" of the local air traffic situation showing all aircraft and their position, identifying information, track/heading and velocity.

  - Developed and maintained a TCP/IP-based data distribution server application with content-based filtering.

  - Developed software to control an ADS-B radio receiver, decode data, and serve to clients.

  - Developed data analysis software for the FAAs Runway Status Lights program, to aid a study tasked with determining where and how many new runway status lights should be installed at several dozen of the nation's top airports. Also developed functional verification software to verify the correct operation of the Runway Status Lights system.

  - Maintained and administered multiple distributed TCP/IP networks (including Ethernet-based LANs and T1 circuits) and network equipment (Cisco routers and managed switches) linking remote sites to supply realtime airport surveillance data to clients.

Jan 1997 – Oct 2001 **Foliage Software Systems, Inc.** – Burlington, MA

http://www.foliage.com

Job Title Senior Software Engineer

Technologies Used • C/C++ • STL • UML • XML • HTML • ASP • JavaScript • CORBA • Windows/MFC/COM • Solaris • Motif • VxWorks • MercuryOS • TI C3x/C4x DSP • MS SourceSafe • RCS • IMAKE

Details Participated in multiple client projects encompassing extensive software design and development work on a broad array of technologies and products, both individually and as a team member, with a strong emphasis on object-oriented methodologies. Summary of projects worked on:

- Enhanced the control software for a commercial ion implantation system. Enhancements ranged from improvements to the advanced robotic wafer handling system which significantly reduced accidental wafer breakage (potentially saving end-users thousands of dollars per incident) to user interface extensions that enabled secure remote access to recipe editing facilities (and others), allowing end-users to optimize their access to system resources. Developed software in C for a VxWorks and SunOS/Motif environment.
- Participated in a proof-of-concept project to design and develop enhancements to a set of deployed commercial electron beam lithography systems. Used an object-oriented design process and developed software using C++ and COM to simultaneously run on Windows NT, Solaris, and a bank of parallel processors running MercuryOS.
- Designed and developed an object-oriented, distributed control system framework for a commercial electron beam lithography system. The system manufacturer used the framework to develop their next-generation lithography system in C++, saving them a great deal of development time and enabling them to rescue a badly slipped development schedule to deliver their new product on time and on budget. Used an object-oriented design process, including UML modeling, and implemented the design in C++ using STL and CORBA. Developed a multi-platform build tool using IMAKE that supported Solaris, Windows NT, and VxWorks.
- Designed and developed embedded avionics software, including a primary flight display and a navigation display, for a small aircraft. Software was developed to DO-178B Level B guidelines and received FAA certification. Utilized an object-oriented design process, including extensive UML modeling. Designed and developed an OO framework in a C environment which provided C++-like functionality, including inheritance and true polymorphism, and allowed the team to quickly translate the OO design into C code for a timely delivery to the client. Applications ran in a custom embedded environment on a TI C3x / C4x DSP, cross-compiled and emulated from Windows NT.
- Designed and developed medical software for a blood testing application. The software was considered a "process control" application because it only partially controlled the hardware environment while primarily intending to interactively guide a human operator through all the steps of the testing process, as well as computing and storing the final results. Developed a custom GUI using Visual C++ and MFC to run in a Windows 2000 environment. Also developed several major system components.
- Enhanced the control software for an advanced digital jukebox. Developed software in C++ and JavaScript/ASP/DHTML, for a Windows 2000 and COM+ environment, using ATL, STL, and MFC. Designed and developed an event logging system which was implemented as a COM component, wrote log files in XML format to enhance machine readability, and accepted queries from Active Server Pages (ASP) to which it responded with XML-formatted event data. Designed and developed a simulator for a custom HTTP server which acted as the interface to a remote, proprietary song database used to automatically populate jukeboxes in the field (restaurants, bars, etc.). Enhanced the XML processing components of the jukebox configuration management sub-system to use multiple, overlaid XML files that allowed new factory configuration data to be introduced into an end-user installation while preserving site-specific settings when possible. Made enhancements to the web browser-based GUI.

Oct 1994 – Dec 1996 **National Datacomputer, Inc.** – Billerica, MA

Job Title Senior Software Engineer

Technologies Used • C/C++ • 80x86 assembler • MS-DOS • DPMI • Windows • Novell NetWare • RCS

Details Developed new and maintained existing software in C and assembler for a PC-based, hand-held computer, with a concentration on user interface facilities. Designed and developed an advanced full-screen editor for proprietary database files, written in C++ for an MS-DOS/DPMI environment, which enabled customer support personnel to work more efficiently and reliably. Designed and maintained adaptable software build procedures for an MS-DOS environment.

Jul 1992 – Oct 1994 **Practice Management Systems, Inc.** – Needham, MA

Job Title Software Engineer

Technologies Used • BASIC • C/C++ • 80x86 assembler • MS-DOS • Novell NetWare

Details Developed new and maintained existing software for electronic medical insurance claim submission and medical billing. Wrote MS-DOS programs using MS-BASIC PDS and Borland C/C++, including network support for Novell networks using Novell NetWare API. Developed user interfaces, function libraries, and toolbox interfaces. Developed telecommunication software for claims submissions over proprietary networks using Borland C/C++ and Greenleaf communications library, and telecommunication scripts using ProComm Plus and other communications software packages. Wrote numerous tools and utilities including a note-taker application, used by telephone operators, that is capable of routing messages over a Novell Network to any printer, allowing customer service calls to be handled more efficiently.

Mar 1992 – May 1992 **Interactive MicroSystems, Inc.** – Salem, NH

Job Title Programmer/Consultant (short-term contract)

Technologies Used • C • 68000 assembler • Commodore Amiga

Details Developed new and enhanced existing video editing and controller software to control video editing decks and process “edit-decision lists”. Developed a software library to load and display computer animations and still pictures transparently from multiple graphics storage formats. Integrated software with an external, commercial scripting language (Arexx). Software was written using C and 68000 assembler for the Commodore Amiga.

Jun 1989 – Mar 1992 **Theta Systems, Inc.** – Woburn, MA

Job Title Software Engineer

Technologies Used • C • 68000 assembler • Commodore Amiga • CDTV • VAX/VMS • VAX MACRO

Details Participated in multiple projects:

- Designed and developed CD-ROM-based, interactive multimedia software in C and 68000 assembler for the Commodore Amiga and Commodore CDTV, including the CDTV Welcome Disk, a product which was distributed worldwide with the CDTV player. Assumed responsibility for the Welcome Disc project under crisis circumstances: numerous outstanding changes and a very limited time budget. Converted the application into a script-based multimedia engine which allowed all changes to be completed before an important deadline. This also made the software usable for several other applications, saving the client a great deal of additional development effort.
- Co-authored the official developer’s guidelines for CDTV, defining the proper behavior of CDTV applications and the standard user interface look and feel for the CDTV development community, accompanied by example programs and code libraries to demonstrate the principals described within.
- Designed and developed telecommunications software using C for the Commodore Amiga and VAX/VMS.
- Designed and developed IPC software for VAX/VMS in C and VAX MACRO.
- Designed and developed object-oriented drawing software using C for the Commodore Amiga.
- Researched and developed OOP methodologies as well as software coding techniques to make programs easily portable between different operating environments such as VMS, Amiga, and MS-DOS.
- Received some exposure to Macintosh and UNIX.

Oct 1988 – May 1989 **Viking Technologies, Inc.** – Middletown, RI

Job Title Managing Editor/Programmer

Technologies Used • C • BASIC • 6502 assembler • Commodore 64/128/Amiga • MS-DOS

Details Produced the Commodore 64/128 edition of UpTime a monthly, computer-oriented, diskette-based periodical. Responsibilities included: planning future issues; making editorial assignments; writing and editing articles and programs; working with an artist to procure artwork for publication; assembling the final product; managing a network of beta testers; and maintaining a monthly budget for all activities. Assisted in the startup of an Amiga edition and programmed for the Amiga and the IBM-PC editions. Developed hypertext authoring and presentation software on the IBM-PC.

Jan 1987 – Nov 1989 **Applied Innovations, Inc.** – Wakefield, RI

Job Title: Software Engineer (part time from Oct 1988 through Nov 1989)

Technologies Used: • BASIC • C • Turbo Pascal • MS-DOS • PVCS

Details: Developed new and maintained existing MS-DOS software for psychological testing and medical billing using Turbo C, Turbo Pascal, 80x86 assembler, and QuickBASIC. Conceived, proposed, designed and developed a new system in Turbo Pascal to generate printed medical insurance claim forms utilizing a full-screen editor and separate printing software. This software drastically expedited support for new claim forms, generating notable customer satisfaction and significant new income for the company. Managed company-wide software version control using PVCS. Continued working part-time on a consulting basis for one year after departing the company in November 1988.

Other Experience

2003 – present **Various Open Source Projects**

Participated in and submitted code to several Open Source projects, including some initiated by myself.

- *Personal Projects* web page:
  http://www.interthingy.net/ray/projects/
- *Stomper* virtual pedalboard/live effects processor for guitar):
  https://sourceforge.net/projects/stomper/
- *Argent* (RSS feed agent):
  https://sourceforge.net/projects/argent/
- *Rex* (text processing utility):
  https://sourceforge.net/projects/rex-text-tool/
- *Rockbox* (alternate DAP firmware):
  http://www.rockbox.org/
- *Boost.Build* (software build system):
  http://www.boost.org/boost-build2/
- *Code::Blocks* (cross-platform C++ IDE):
  http://www.codeblocks.org/
- *Codelite* (cross-platform C++ IDE):
  http://codelite.org/
- *Kismet Qt/E* (Kismet front-end for the Sharp Zaurus PDA):
  https://sourceforge.net/projects/kismet-qte/
- *Guarddog* (Linux firewall configuration utility):
  http://www.simonzone.com/software/guarddog/

Mar 1984 – Mar 1987 **M.E.M.O. Computer User's Group** – Fall River, MA

Position: Librarian (elected 3 terms)

Technologies Used: • BASIC • 6502 assembler • Commodore 64

Produced a monthly club diskette containing public domain programs for the Commodore 64 computer and presented it at public meetings. Organized and taught club-sponsored programming classes in BASIC and 6502 assembly language programming.

Education

1981 – 1982 **Bristol Community College** – Fall River, MA

Area of Study: Associates Degree in Electrical Engineering (incomplete)

1980 – 1981	**Bishop Connolly High School** – Fall River, MA
Area of Study	Introduction to BASIC programming

References

Available upon request.

Updated: 2018-11-20