Bart Massey

Email: bart@cs.pdx.edu
Telephone: 503.725.5393

Open Source

Projects

Bart is currently leading a variety of projects that are both open source development efforts and testbeds for ideas and methods in open source software engineering. This document describes some of the most significant of these projects.

1 PSAS Rocket Avionics

Over the past few years, the Portland State Aerospace Society (PSAS) has constructed a high-altitude amateur sounding rocket with perhaps the most advanced low-cost rocket avionics system in the world. This avionics package integrates home-built open-hardware components such as a 6-axis inertial measurement unit, a GPS, 802.11b telemetry, amateur television, and a variety of other functions using an on-board Linux flight computer; the resulting system provides sophisticated inertial navigation capabilities, with guidance planned. In support of the avionics is a variety of ground control and telemetry systems. All of these systems have a major software component.

1.1 Project Team

Team Leads: Andrew Greenberg, MS ECE student. Jamey Sharp, undergraduate (Software Team). Brian O’Neel, MS ME student (Airframe and Propulsion Team). Students: The PSAS team. See the website for a full listing.

1.2 Status

Software and hardware for PSAS LV2 is complete. A launch is planned for late April or early May. A generous award from IBM has provided funds for upgrading the Linux flight computer to a higher-performance PowerPC platform: this integration is underway.

All software and hardware is freely available under the GPL. One of the project goals is low cost: high schools and colleges can and have replicated PSAS work. For much more information, see URL http://psas.pdx.edu.

1.3 bibliography

James Perkins, Andrew Greenberg, Jamey Sharp, David Cassard and Bart Massey. Free Software and High-Power Rocketry: The Portland State Aerospace Society. In Proc. 2003 Usenix Annual Technical Conference, Freenix Track, San Antonio, TX, June 2003. URL http://psas.pdx.edu/psas/usenix_2003/psas.pdf.

2 Launch Vehicle Simulator

In order to validate the PSAS launch vehicle, a launch vehicle simulator was recently constructed. The first prototype of this simulator was constructed in Nickle, and used to validate the flight computer software for LV2. Subsequent work produced a much more sophisticated discrete event simulator in C++ comprising about 35KLOC. Current work on the simulator involves completing a more detailed three-dimensional physics model. This simulator should be useful for other discrete-event physics simulations in rocketry and elsewhere.

2.1 Project Team

Team Lead: David Allen, MS student

2.2 Status

The current simulator software is available under the GPL as part of the PSAS project. As the simulator progresses, it will be separated for further development

3 The XCB X Window System Project

The X Window System is the basis of the premier desktop environments for UNIX systems. The XCB project provides a lightweight, high-quality replacement for the lowest X client layer, Xlib. The Xlib/XCB project provides Xlib compatibility atop XCB. This work also provides a testbed for exercises in software engineering and formal methods. Xlib/XCB is expected to be adopted as a replacement for Xlib by freedesktop.org/X.org in the medium term.

3.1 Project Team

Team Lead: Jamey Sharp,undergraduate
undergraduate Students: Josh Triplett, undergraduate

3.2 Status

XCB has been freely available under an open source license since 2002. The team has currently added several external committers, and is making progress toward the first production-quality release of Xlib/XCB. Patches to support Xlib/XCB have already been accepted into the freedesktop.org Xlib tree. The work of Robert Bauer < rtbauer@us.ibm.com> , IBM Rational Systems Division, in applying formal methods during XCB development is also gratefully acknowledged. For more information, see http://freedesktop.org/wiki/Software_2fxcb.

3.3 Bibliography

Jamey Sharp. How Xlib is Implemented (And What We’re Doing About It). In Proc. 2004 Usenix Annual Technical Conference, Freenix Track, Boston, MA, June 2004. URL http://www.usenix.org/events/ usenix04/tech/freenix/sharp.html. Jamey Sharp and Bart Massey. XCL: An Xlib Compatibility Layer For XCB. In Proc. 2002 Usenix Annual Technical Conference, Freenix Track, Monterey, CA, June 2002. URL http://xcb.cs.pdx.edu/usenix-xcl.pdf. Bart Massey and Jamey Sharp. XCB: An X Protocol C Binding. In Proc. 2001 XFree86 Technical Conference, Oakland, CA, November 2001. USENIX. URL http: //xcb.cs.pdx.edu/papers/xcb.pdf. Bart Massey and Robert Bauer. X Meets Z: Verifying Correctness In The Presence Of POSIX Threads. In Proc. 2002 Usenix Annual Technical Conference, Freenix Track, Monterey, CA, June 2002. URL http://xcb.cs.pdx.edu/usenix-zxcb.pdf.

4 The XML-XCB X Protocol Description

Work on the XCB project highlighted the need for a machine-readable description of the X Window System protocol. Initial work was done using an M4 meta-language. This was effective for generating XCB code, but diffcult to work with and re-use. The XML-XCB project comprises an XML Schema for X protocol description, a conforming XML description of the X protocol, and XSLT stylesheets and related processing tools for transforming the protocol description into useful forms. Substantial portions of XCB and Xlib/XCB are auto-generated from the XML-XCB protocol description. Immediate plans include an X protocol analysis plugin for Ethereal that is also auto-generated from XML-XCB.

4.1 Project Team

Team Lead: Josh Triplett, undergraduate

4.2 Status

XML-XCB is freely available under an open source license. Work on the Ethereal plugin is nearly complete. For more information, see http://freedesktop.org/wiki/Software_2fxcb

5 The Nickle Programming Language

The Nickle programming language features C-like syntax surface semantics as well as a variety of powerful imperative and functional features, including arbitrary precision integer, rational, and floating-point arithmetic. The environment is useful as a desk calculator for engineers and as a platform for algorithm development. It is also a tool for exploring ideas in programming languages and software engineering.

5.1 Project Team

Team Leads: Keith Packard < keithp@keithp.com> , HP Cambridge Research Laboratory, and Bart Massey < bart@cs.pdx.edu>
Students: James LaMar and Emma Kuo, undergraduates.

5.2 Status

Nickle has been freely available in source form by request since 1988. Since 2000, it has been widely distributed using standard open source channels. For more information, see http://nickle.org.

5.3 Bibliography

Bart Massey and Keith Packard. Nickle: Language Principles and Pragmatics. In Proc. 2001 Usenix Annual Technical Conference, Freenix Track, Boston, MA, June 2001. URL http://www.nickle.org/usenix-nickle. pdf. Bart Massey. Algorithm Prototyping and the Nickle Prototyping Environment. In International Conference on Software Engineering, Portland, OR, May 2003. Submitted. Bart Massey. “Nickle’s polite type system”, October 2004. Invited Presentation.

6 MINT

MINT is a Nickle-based language translation tool. The basis of MINT is an integrated lexer generator and Canonical LR(1) parser generator. The principal novelty is the production of lightweight language-independent representations of translators, such that abstract syntax trees can be produced by a parser written in a target language of choice. MINT was conceived as a Nickle bootstrap tool, but has also already proven useful for rapid prototyping of "little languages" with minimal effort.

6.1 Project Team

Team Lead: James LaMar, undergraduate.
Students: James LaMar and Emma Kuo, undergraduates.

6.2 Status

A preliminary MINT implementation is currently available under an open source license. This project is still new, and work is ongoing. For more information, see http://nickle.org.

6.3 Bibliography

Emma Kuo, James LaMar, Bart Massey and Keith Packard. "MINT Is a Nickle Translator: Generating cross-lingual recognizers", December 2004. Preliminary draft: not for redistribution. URL http://nexp.cs.pdx. edu/mint-draft.pdf.

Concluding Comments

Bart plans to continue to participate at a high level in open source software development and related activities both locally and internationally. With the aid of partners like IBM, OSDL, freedesktop.org, and Usenix, he expects to accomplish a great deal in the short term.

Bart Massey is an Assistant Professor of Computer Science at Portland State University, and a faculty member in the Oregon Member of Software Engineering Program. His chief research interests are in open source software engineering and its applications. Bart works in a variety of domains, including rocketry, artificial intelligence, the X desktop, and in the application of formal methods to open source development.

Bart has been heavily involved in the open source software development community for many years. He recently co-chaired the Freenix Track of the Usenix Annual Technical Conference. Bart was instrumental in obtaining an Intel grant to bring a Linux Laboratory to PSU. His coursework includes a regularly-taught Summer laboratory class on open source software development (http://wiki.cs.pdx.edu/ossclass). Bart will co-teach a class on Linux Kernel and Device Driver Construction this Spring.

Bart worked as a Software Engineer III at Tektronix Inc. prior to his doctoral work in Computer Science. He received his doctorate from the University of Oregon in 1999, specializing in Artificial Intelligence at the Computational Intelligence Research Laboratory.