Version Control: A Case Study in the Challenges and Opportunities for Open Source Software Development
(Position Paper for 2nd Workshop on Open Source Software Engineering)

Mark C. Chu-Carroll, David Shields and Jim Wright
{mcc,shields,jwright}@watson.ibm.com
IBM T. J. Watson Research Center
19 Skyline Drive, Hawthorne, NY 10522

Abstract

The growth of the worldwide open source development effort, driven in part by the recent entrance of large corporations into the open source arena, offers new opportunities to improve the software engineering tools available for that effort. Indeed, the increasing difficulty of managing large open source projects, as well as that of integrating related efforts into new programming environments, represents a challenge that must be met if the rapid growth of open source software is to continue. This position paper addresses these issues in the context of software version control.

Version Control and the Linux Kernel

The Linux kernel, perhaps the most crucial piece of open source software, represents an interesting example in that the kernel developers use a primitive form of version control. Patch files are exchanged by email and posted to web sites, and ultimately coordinated by a single individual, Linus Torvalds, aided by a cohort of trusted assistants [1].

Though the kernel has achieved a state of remarkable quality and stability, there are many variants of the kernel source tree [2] , and problems coordinating updates and the addition of new features have been reported, as noted in a discussion of Virtual Memory Managers [3] . Torvalds has recognized these problems [4] , and recently reported that he is using the Bitkeeper version control system [5, 6, 7] .

Compiling the Linux kernel itself presents a complicated configuration problem, recently addressed by Eric Raymond, the author of CML2 [8]," a configuration system ... that handles build-option selection for Linux kernels" that is "scheduled to be integrated into the Linux kernel source tree between 2.5.1 and 2.5.2." (Text in quotes, here and later on, is from the referenced web sites.) CML2 is written in Python and, according to the CML2 Announcement [9]: "For those of you who grumbled about adding Python to the build-tools set, Linux has uttered a ukase: CML2's reliance on Python is not an issue" (See also [10]).

New Resources for Building Programming Tools

The toolkit of open source developers has been expanding with recent additions of a variety of new tools. Within the open source commmunity, development is no longer limited to C, shell scripts, and the standard unix tools such as awk, sed and perl . Programmers can now use a variety of programming languages and tools, and the use of highly flexible interpreted languages and virtual machine platforms has become an acceptable practice. The maturity and acceptability of such tools is demonstrated by the the imminent acceptance of a tool written in Python by the Linux kernel community, Java in the Apache community [11], and the use C# and the CLR for use in GNOME applications [12].

More active corporate involvement in the open source arena has produced tools such as Java [13]. Though precise figures are not available, it seems safe to say that IBM and Sun have jointly invested well over a hundred million dollars bringing Java to its current state, and with mature versions for Linux available from IBM and Sun, Java is a major addition to the tool builder's arsenal. Indeed, several communities favoring Java's use have emerged, including the Jakarta Apache Project, which has produced many useful programs, and Tigris [14], a "mid-sized open source community focused on building better tools for collaborative software development."

It is also worth noting that many open source developers are increasingly willing to work in mixed-license environments when doing so allows them to use significantly better tools. Thus, many developers have adopted the Perforce SCM system [15] due to its practice of granting free licenses for open source development, and the kernel developers are using BitKeeper despite its hybrid license, which is part open source, part proprietary.

New Version Control Systems

The availability of new tools and new technologies has presented an opportunity to re-engineer some of the common software tools. This, coupled with the increasing recognition of the need for better tools to manage the increasing complexity of open source development, has resulted in renewed interest in areas such as version control that have not been the subject of active research. For example, CVS, the dominant open source version control system, has seen little change in the last decade. That's a long time in the software world -- CVS is "older" than the Internet, HTML, Linux, Java and XML -- and there are several projects working on new approaches to version control:

- Arch [16], a "revision control system that features distributed repositories, whole-tree patch sets, ... and easy integration into automated software engineering processes."
- BitKeeper [5], a "powerful distributed source management system."
- Subversion [17], a "version control system that is a compelling replacement for CVS in the open source community."
- Stellation [18], our project from IBM Research, which aims to develop programming environments for collaborative development. Work to date has focused on developing the necessary infrastructure for long-term research goals that include work on fine granularity (working on source fragments smaller than a single source file), dynamic program organization, and repository replication. The first use of the infrastructure has been the development of a new version control system, Svc, that provides many of the expected functions, as well as novel features including the use of a relational database as the backend, a "save/restore" facility to manage workspace states while preparing a new project revision, and a robust project-level versioning which maintains change histories for moved and renamed artifacts.

As an example of the power of the open source tools that are available, we can compile the 30,000 plus lines of our Java source using Jikes [19] and Ant [20], a build tool from Jakarta, in under five seconds . We also use Log4j [21], a fast and flexible logging library, and ORO [22], consisting of Java classes that provide Perl5 compatible regular expressions, from the Jakarta Project. All our configuration files, as well as the messages used by our distribution layer, are in XML, manipulated using the JDOM package [23].

Our project uses a relational database for the program repository. Though we use the open source database PostgreSQL [24], our use of JDBC [25] should allow the use of other relational databases. Indeed, the availability of industrial-strength relational databases for Linux from IBM and Oracle that offer unmatched reliability and performance, as well as the possibility of novel information retrieval techniques based on SQL, may well lead to the re-engineering of other development tools.

As a final example, the problem of how best to synchronize data between a laptop and a desktop [26], is similar to that of exchanging files and notifications within a distributed development community. The InterMezzo Project [27] reports that, "InterMezzo is a filtering file system that generates a modification log file which is suitable for use on other hosts. InterSync is a scalable client server system to synchronize InterMezzo filesystems...Intermezzo became part of the Linux Kernel since 2.4.15." The availability of a synchronizing capability within a filesystem offers the possibility of providing new ways to exchange updated files, differences, change sets, and perhaps even build files as part of a version control system.

Version Control For Integrated Development Environments

Increased corporate involvement has also produced the ambitious Java-based development environments Eclipse [28] and NetBeans [29]. The Eclipse Platform [30] "is designed for building integrated development environments (IDE's) that can be used to create applications as diverse as web sites, embedded Java programs. C++ programs, and Enterprise JavaBeans." NetBeans is "modular, standards-based IDE written in Java ... with a wide range of features from JSP development and debugging, to integrated CVS support and beyond...implemented in modules that plug into the NetBeans core [31]." The use of such open source develpment tools dramatically raises the level of tools support possible for open source development.

Eclipse and NetBeans both use the notion of *plug-in* that permits separate development of components that can be integrated into a larger environment. As described in *Inside Eclipse and the WebSphere Studio Family* [32]: "A plug-in can use capabilities (called extension points) that the platform runtime or other plug-ins provide. In turn, a plug-in can optionally include extension points for other plug-ins to use ... The plug-in's compiled Java code includes the implementation of its capabilities and extension points."

Eclipse Version and Configuration Management [33] and NetBeans Version Control [34] both currently support the use of CVS for version control. CVS and Subversion are written in C, while arch is written primarily in shell scripts with some C code. Though the plug-in model permits the integration of tools not written in Java, the availability of a version control system such as Svc written in Java offers the possibility of tighter integration of the version control process with the development environment. This ability to extend the platform allows these tools to become more than just the sum of their parts: a sophisticated IDE integrated with more advanced version control is significantly better than either alone. Such a combination through plugins represents a major step forward in the level of tooling available for open source developers.

Though a tool may be more effective when used within a programming environment, we also think it useful to provide a command-line form where possible, to permit use outside a specific environment.

The "competition" between Eclipse (sponsored by IBM) and NetBeans (sponsored by Sun) illustrates a novel aspect of open source. Were these projects available only as closed source, the user would have to choose between them, while their availability in open source form permits, subject to license restrictions, the "harvesting" by the open source community of the best features of each. For an example of such sharing, consider Mozilla [35], "an open-source web browser designed for standards compliance, performance and portability." Some of the Mozilla technology has been used by other projects, such as Galeon [36], a Gnome [37] web browser based on Gecko, Mozilla's embeddable layout engine.

The use of plug-ins, and the need for coordination of related components into software aggregates like Eclipse and Netbeans, is another instance of the problem of package management that is becoming more of an issue in open source development. While this is not a new issue, as witness the discussion of package management for emacs [38], it may become a fruitful area for more active research.

Conclusion

While the increased growth and complexity of open source development have exposed a number of problems, there are interesting opportunities for applying newly available tools and technologies. The recent activity in version control systems is but one sign of the work underway by the open source community to address these problems. Hopefully, further work on these problems will lead to even better software development tools and practices, supporting more effective management of large, complex open source projects, and easing integration with new programming environments.

References:

[1] Nicholas Petreley. Kernel Source Merging 101. Available Online at <http://linuxworld.com/site-stories/2002/0211.merge.html>

[2] Moshe Bar. A Forest of Kernel Trees. Available Online at <http://www.byte.com/servinglinux/2002/02/>

[3] Moshe Bar. Linux Kernel Pillow Talk. Available Online at <http://www.byte.com/servinglinux/2002/02/>

[4] Joe Barr. Linus tries to make himself scale. Available Online at <http://linuxworld.com/site-stories/2002/0211.scale.html>

[5] BitKeeper Home Page. Available Online at <http://www.bitkeeper.com>

[6] Linus Torvalds: Linux-2.5.4-pre1 - bitkeeper testing. Available Online at <http://linuxtoday.com/news_story.php3?ltsn=2002-02-06-006-20-NW-KN-DV>

[7] Jeff Garzik; Doing the BK Thing, Penguin Style (Notes on Bitkeeper for Kernel Developers). Available Online at <http://news.linuxprogramming.com/news_story.php3?ltsn=2002-02-21-001-06-DT-HT>

[8] Eric S. Raymond. The CML2 Resources Page. Available Online at <http://tuxedo.org/~esr/cml2/>

[9] Eric S. Raymond. CML2 Announcement. Available Online at <http://tuxedo.org/~esr/cml2/ANNOUNCEMENT>

[10] Jeremy Andrews. Linux: CML2, ESR and the LKML. Available Online at <http://kerneltrap.org/node.php?id=17>

[11] The Jakarta Project Home Page. Available Online at <http://jakarta.apache.org>

[12] DotGNU Project Home Page. Available Online at &lthttp://www.dotgnu.org/>

[13] Java Home Page. Available Online at <http://java.sun.com>. Java is a trademark of Sun Microsystems.

[14] Tigris.org Home Page. Available Online at <http://tigris.org>

[15] Perforce Home Page, Licensing/Pricing. Available Online at <http://www.perforce.com/perforce/price.html>

[16] Arch Project Home Page. Available Online at <http://www.regexps.com/#arch>

[17] Subversion Project Home Page. Available Online at <http://subversion.tigris.org/>

[18] Stellation Project Home Page. Available Online at <http://domino.research.ibm.com/synedra/synedra.nsf>

[19] Jikes Project Home Page. Available Online at <http:://oss.software.ibm.com/developerworks/opensource/jikes>

[20] Ant Project Home Page. Available Online at <http://jakarta.apache.org/ant>

[21] Log4 Project Home Page. Available Online at <http://jakarta.apache.org/log4j>

[22] ORO Project Home Page. Available Online at <http://jakarta.apache.org/>

[23] JDOM Project Home Page. Available Online at <http://jdom.org>

[24] PostgreSQL Project Home Page. Available Online at <http://www.postgresql.org>

[25] JDBC Data Access API. Available Online at <http://java.sun.com/products/jdbc>. JDBC is a Trademark of Sun Microsystems.

[26] Moshe Bar. Keeping in Sync. Available Online at <http://www.byte.com/servinglinux/2002/01/>

[27] InterMezzo Project Home Page. Available Online at <http://www.inter-mezzo.org>InterMezzo>

[28] Eclipse Project Home Page. Available Online at <http://www.eclipse.org>

[29] Netbeans Project Home Page. Available Online at <http://www.netbeans.org>

[30] Eclipse Platform Technical Overview. Available Online at <http://www.eclipse.org/whitepapers/eclipse-overview.pdf>

[31] The NetBeans Platform and the NetBeans IDE. Available Online at <http://www.netbeans.org/about.html>

[32] Paule Conte. Inside Eclipse and the WebSphere Studio Family. Available Online at <http://www.e-

promag.com/eparchive/index.cfm?fuseaction=viewarticle&ContentID=1708>

[33] Eclipse Version and Configuration Management. Available Online at <http://dev.eclipse.org/viewcvs/index.cgi/~checkout~/platform-vcm-home/main.html>

[34] NetBeans Version Control. Available Online at <http://versioncontrol.netbeans.org>

[35] Mozilla Project Home Page. Available Online at <http://mozilla.org>

[36] Galeon Project Home Page. Available Online at <http://galeon.sourceforge.net>

[37] Gnome Project Home Page. Available Online at <http://gnome.org>

[38] "XEmacs vs. GNU Emacs". Available Online at <http://www.xemacs.org/About/XEmacsVsGNUemacs.html>