Software Engineering Research in the Bazaar

Ahmed E. Hassan, Michael W. Godfrey, and Richard C. Holt
Software Architecture Group (SWAG)
Department of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1
CANADA
{aeehassa, migod, holt}@plg.uwaterloo.ca

ABSTRACT

During the last five years, our research group has studied the architecture and evolution of several large open source systems — including Linux, GCC, VIM, Mozilla, and Apache — and we have found that open source software systems often exhibit interesting differences when compared to similar commercially-developed systems. Our investigations of these systems have involved the creation of software architecture models, software architecture repair, the creation of a reference architecture for web servers, the study of evolution and growth of open source systems, and the modelling of architectural properties of systems that are apparent only at build time.

Keywords
Open Source Software, Software Architecture, Software Engineering Research

1 INTRODUCTION

Software engineering research studies the structure and development of large software systems. Many researchers have opted to use small academic software systems to study and to validate their research. The extrapolation of useful findings from 100 LOC system to a 1 MLOC system are rarely valid.

Fortunately, the open source community provided us the opportunity to study large non trivial software systems. Furthermore, the publication of our work is not constrained by NDAs usually required by commercial companies. As we studied many open source systems, we discovered that open source systems exhibit many interesting traits that are not found in previously studied commercial systems. In the following sections, we detail several areas of active research in our group and our findings.

2 SOFTWARE ARCHITECTURE RECOVERY AND REPAIR

Many software systems do not have a documented system architecture. The maintenance and the understanding of these large and complex system is very difficult because of the lack of up-to-date documentation. To gain a better understanding of a software system, we recover its software architecture [4, 11] from the system's source code.

We recovered the architecture of many open source systems such as the Linux kernel [3], the Mozilla browser [5], the Apache web server [7], and the VIM editor [14]. The recovered architecture is browse-able to permit developers to interact with it, [10] shows an example for the Linux kernel.

Furthermore, we found that many differences existed between the as-built architecture and the as-designed architecture for these systems. This architectural drift, which can impede program understanding, is especially pronounced in open source systems, where many developers work in isolation on distinct features with little coordination. We attempted to reconcile (*repair*) the differences between the as-built and the as-designed architecture for VIM and Linux [15].

3 REFERENCE ARCHITECTURE

A reference architecture for a domain defines the fundamental components of the domain and the relations between them. A reference architecture is very beneficial for reuse, maintenance and new development. Mature domains such as compilers and operating systems have well-known reference architectures. For example, a compiler is understood to have a scanner, parser, semantic analyzer and a code generator subsystem [12].

In [7], we derived a reference architecture for web servers. To derive it, we recovered the architecture of three large open source web servers (Jigsaw [8], Apache [2], and AOLServer [1]). We studied the architecture of all three web servers to find communalities which we abstracted into a reference architecture for web servers. For commercial software systems, we would not have been able to acquire three different implementations of a web server developed by three competing companies.

4 SOFTWARE EVOLUTION

Most studies of software evolution were performed on systems developed within a single organization. We choose to examine the evolution of the Linux kernel because of the large size of its code base and because its development is not as tightly managed like commercial software systems.

Expecting to find the growth rate of the kernel to slow down as its size and complexity increased. we were surprised to find out that the kernel is growing at a super linear rate [6]. Further investigations showed that architecture of the kernel and its open source development model were the main factors that explained its growth rate. Currently, we are investigating the evolution of the GCC compiler.

5 BUILD-TIME VIEW

During our study of open source systems, we found that they exhibit interesting and complex build-time properties that aren't explicitly addressed by traditional architecture views [9, 13].

Open source systems are, in general, designed to be as portable as possible. Rather than create a separate source distribution for each platform, the commonalities are abstracted into a single distribution and configuration tools are used to aid in building the system. A build-time view records this complex build structure [16].

6 CONCLUSION

Open source systems provide a great opportunity for researchers to study real life large software systems. We recovered and repaired the architecture of many open source systems such as Linux and VIM. We used these systems to derive a reference architecture for the web server domain. In addition, we examined the evolution of the Linux kernel and were surprised of its fast growth rate. Furthermore, we proposed a new architecture view to address the complex build structure of these systems. We continue our investigations of open source systems and encourage others to join us.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the significant contributions of many graduate students at the University of Waterloo, particularly current and former members of the SWAG research group. We are also grateful to members of the open source community who have given freely of their time in helping us to understand their systems.

REFERENCES

[1] America Online's Open-Source Web Server Home Page. Available online at `<http://www.aolserver.com/>`

[2] The Apache HTTP Server Project Home Page. Available online at `<http://httpd.apache.org/>`

[3] I. T. Bowman, R. C. Holt, and N. V. Brewster. Linux as a Case Study: Its Extracted Software Architecture. In *IEEE 21st International Conference on Software Engineering*, Los Angeles, USA, May 1999.

[4] D. Garlan and M. Shaw. An Introduction to Software Architecture. In V. Ambriola and G. Tortora, editors, *Advances in Software Engineering and Knowledge Engineering*, pages 1–39, Singapore, 1993. World Scientific Publishing Company.

[5] M. W. Godfrey and E. H. S. Lee. Secrets from the Monster: Extracting Mozilla's Software Architecture. In *Proceedings of the Second International Symposium on Constructing Software Engineering Tools*, pages 15–23, Limerick, Ireland, June 2000.

[6] M. W. Godfrey and Q. Tu. Evolution in Open Source Software: A Case Study. In *Proceedings of the 2000 International. Conference on Software Maintenance*, San Jose, California, Oct. 2000.

[7] A. E. Hassan and R. C. Holt. A Reference Architecture for Web Servers. In *7th Working Conference on Reverse Engineering*, Brisbane, Queensland, Australia, Nov. 2000.

[8] Jigsaw - The W3C's Server - Home Page. Available online at `<http://www.w3.org/Jigsaw/>`

[9] P. B. Kruchten. The 4+1 View Model of Architecture. *IEEE Software*, 12(6):42–50, Nov. 1995.

[10] The Software Bookshelf of the Linux Kernel. Available online at `<http://swag.uwaterloo.ca/pbs/examples/linux/>`

[11] D. E. Perry and A. L. Wolf. Foundations for the Study of Software Architecture. *ACM SIGSOFT Software Engineering Notes*, 17(4):40–52, Oct. 1992.

[12] M. Shaw and D. Garlan. *Software Architecture: Perspectives on an Emerging Discipline*. Prentice-Hall, Inc., Upper Saddle River, NJ., USA, 1996.

[13] D. Soni, R. L. Nord, and C. Hofmeister. Software Architecture in Industrial Applications. In *IEEE 17th International Conference on Software Engineering*, 1995.

[14] J. B. Tran. Software Architecture Repair as a Form of Preventive Maintenance. Master's thesis, University of Waterloo, 1999. Available online at `<http://plg.uwaterloo.ca/~j3tran/papers/thesis.ht`

[15] J. B. Tran, M. W. Godfrey, E. H. S. Lee, and R. C. Holt. Architectural Repair of Open Source Software. In *Proceedings of International Workshop on Program Comprehension*, Limerick, Ireland, June 2000.

[16] Q. Tu and M. W. Godfrey. The Build-Time Software Architecture View. In *Submitted to 2001 International Conference on Software Maintenance*, 2001.