
LifeSource: Two CVS Visualizations

Eric Gilbert

University of Illinois at
Urbana-Champaign
201 N. Goodwin Ave.
Urbana, IL 61801
egilber2 -at- uiuc dot edu

Karrie Karahalios

University of Illinois at
Urbana-Champaign
201 N. Goodwin Ave.
Urbana, IL 61801
kkarahal -at- cs dot uiuc dot edu

Abstract

We present *LifeSource*, two visualizations of CVS code repositories, one file-centric and one author-centric. *CodeConnections*, the file-centric visualization, pulls out the overall structure of a large body of interacting code and authors. *CodeConnections* scales well and reveals large trends in the code base at a glance. *CodeSaw*, the author-centric visualization, compares authors' code and project mail contributions over the course of one year. *CodeSaw* reveals trends in authors' lives, exposing peaks and valleys of productivity. Both visualizations uncover deep details about the life of a code project to developers and onlookers.

Keywords

Code visualization, CVS, authorship visualization, Azureus

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous; H1.2 User/Machine Systems: Human Information Processing.

Introduction

A recent study estimates that 1.1 million software developers in North America spend at least some time on open source projects [6]. SourceForge.net hosts many of these projects for free [8]. As a standard offering of their service, SourceForge.net opens the

Copyright is held by the author/owner(s).

CHI 2006, April 22–27, 2006, Montréal, Québec, Canada.

ACM 1-59593-298-4/06/0004.

```

-----
revision 1.429
date: 2005/12/08 05:18:49;
author: nolar; state: Exp;
lines: +6 -0
Clearer firewalled/NAT status
reporting.
-----
revision 1.428
date: 2005/11/22 21:08:26;
author: nolar; state: Exp;
lines: +4 -0
2307 prep.
-----
revision 1.427
date: 2005/11/22 08:03:41;
author: nolar; state: Exp;
lines: +19 -19
Prep.
-----

```

Figure 1. A small snippet of the output of the `cvsl log` command for a file in the Azureus code base.

CVS source control repositories to the public, giving anyone who cares to look an intimate view into the life of a project.

In this paper, we present *LifeSource*, two visualizations that tell the story of a code project. Often, developers on a project only get a myopic view. They may be confined to particular area of the development, focusing hard on a small section. *LifeSource* gives developers and onlookers a chance to step back and look at a code project over time.

CVS stores all of its information in a very raw textual format [Fig. 1]. *LifeSource* digests this textual data into a visual display appropriate for understanding themes. For example, we anticipate that project managers will benefit from *LifeSource's* ability to provide insights into employee potential.

Related Work

In the specific arena of source code visualization, much of the existing work focuses on depicting the nested structure of code. For example, *SeeSys* uses a treemap to display prominent structures within a source code file system [2]. It can provide an overall structure of the Linux kernel. *Augur* presents code repository information in a visual diff-like display [4]. A user digs into a single source file to see the detail, as *Augur* represents it line by line.

Outside of the code visualization arena, *HistoryFlow* depicts the changes by multiple authors to one document [9]. It applied this technique specifically to

Wikipedia entries, although it would have worked for code as well. However, since it focuses on just one text file and gives no consideration to its structure, we created a new visualization.

Authorlines very beautifully shows trends in Usenet postings by a single author [10]. In the case of code repository visualization, our visualization must show interactions between multiple authors.

Implementation

In this paper, we visualize the code repository of Azureus, a Java BitTorrent client [1]. We chose Azureus because, at the time of this writing, it ranks 3rd in download popularity and 1st in development activity on SourceForge.net.

The Azureus CVS repository provides a number of details about each file [Fig. 1]. We built a Perl parser for CVS logs and Java source code. The parser chops the logs and analyzes the source files into their most important features and stores them in a PostgreSQL database.

For visualization, *CodeConnections* uses the yFiles development toolkit [12]. *CodeSaw* uses Processing [7].

CodeConnections

CodeConnections is divided into a top and a bottom pane. The top pane shows the structure of the project at a given time. The bottom gives the user navigational control [Fig. 2].

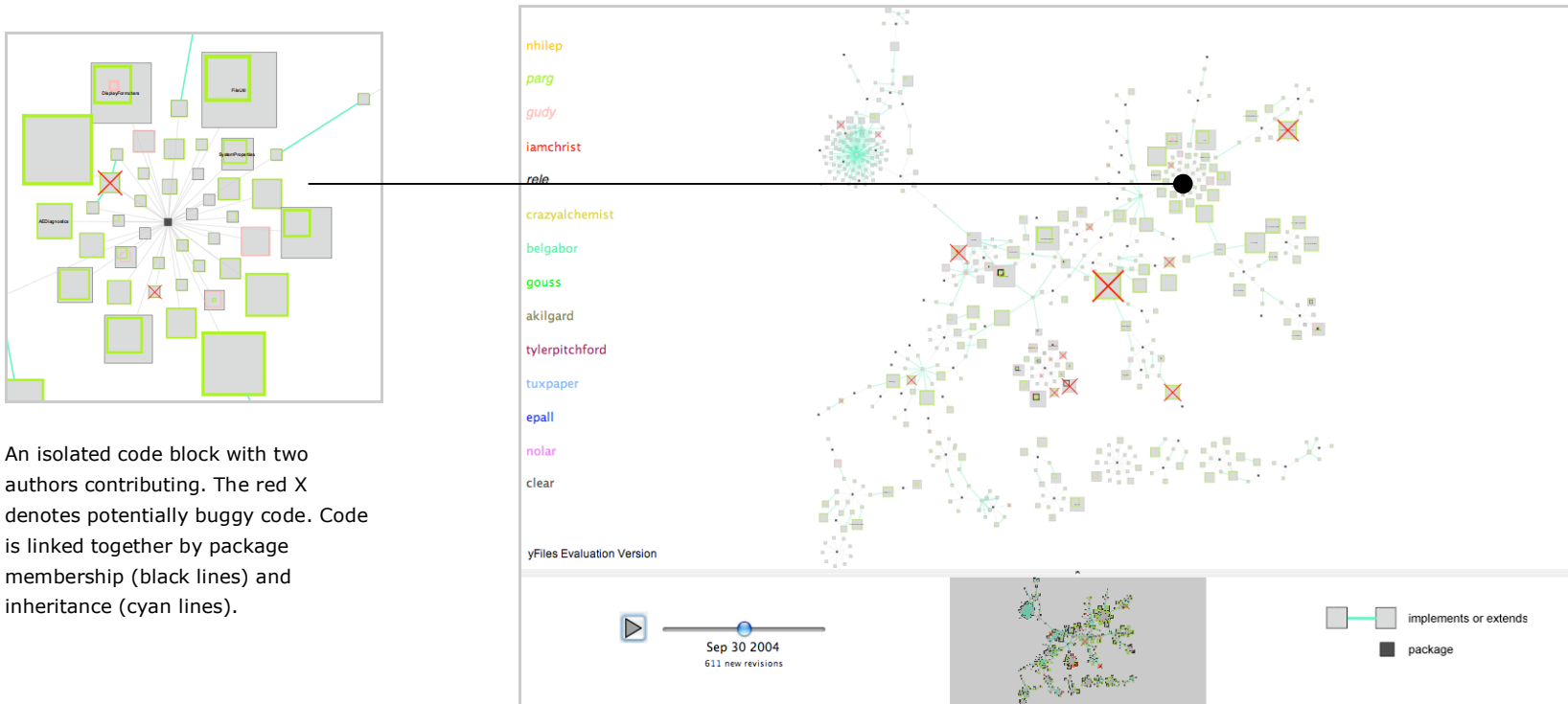


Figure 2. The *CodeConnections* interface with code base and authors in the top pane and navigational control in the bottom pane. We track the contributions of three authors nearly one year into Azureus.

Each square represents a file in the code repository. Squares become larger based on a composite score of file size and file importance. Source files become linked via membership in the same package or by inheritance. In Figure 2, Azureus's 13 authors appear down the left side. Clicking on an author's name causes each file to show the author's relative contribution in the author's color. In addition to the inheritance analysis, the parser determines each file's cyclomatic complexity. A large

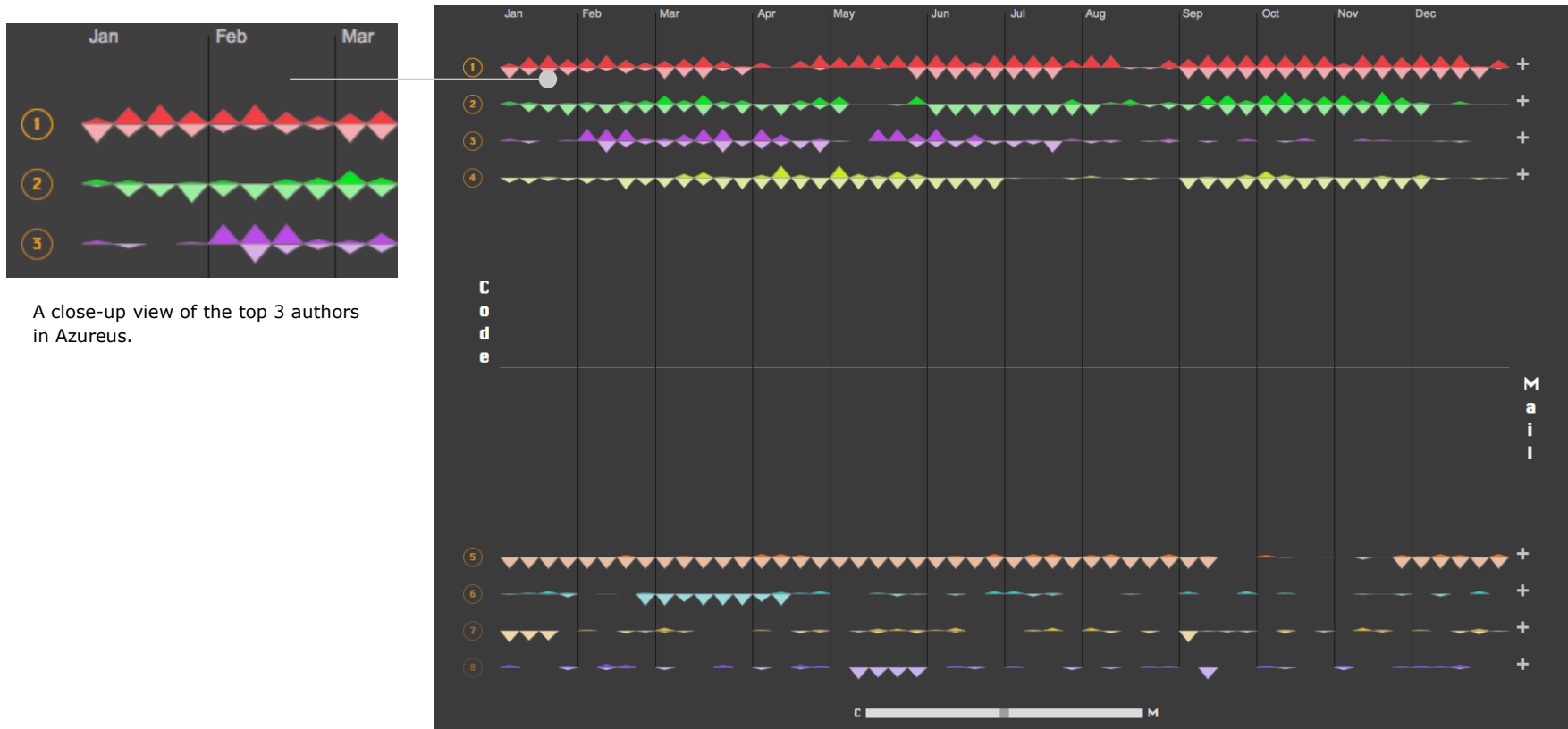
cyclomatic complexity could indicate buggy code [3]. A large red X drawn over the square denotes such files.

The bottom navigation panel allows users to pause and play the visualization in time. The basic time step is two weeks. In the center of this panel, a small representation of the overall structure allows users to navigate to sections of interest.

CodeSaw

CodeSaw shows the contributions of project authors over the course of one year [Fig. 3]. A colored timeline made of triangles represents each author. An author's code contributions sit above each axis (more saturated). Mail contributions to the project mailing list sit below each axis (less saturated). A ranking appears next to each author's timeline. The rank is a weighted

sum of code and mail contributions. Users can set the relative weights using the slider at the bottom of the interface. Clicking on the plus symbol next to any combination of the timelines brings up a more detailed view in the center region [Fig. 4]. Upon clicking, the plus changes to a minus, allowing the user to remove it from the center region when desired.



A close-up view of the top 3 authors in Azureus.

Figure 3. The *CodeSaw* interface showing Azureus's eight most highly ranked authors.

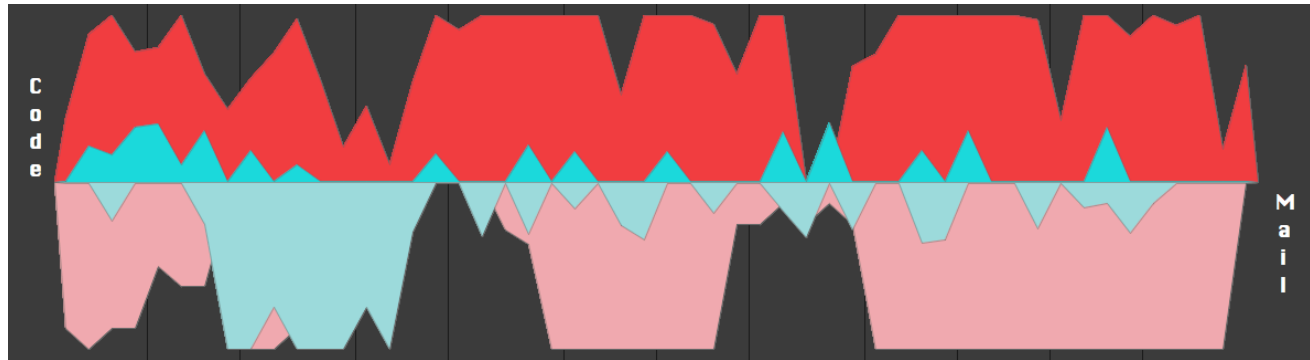


Figure 4. The result of clicking the plus symbol next to the red author (ranked 1st) and the plus symbol next to the cyan author (ranked 6th). The more detailed center view allows the user to bring together any subset of the authors she chooses.

When *CodeSaw* finishes loading the data, each timeline grows up and down from the axis, following the advice from *NameVoyager* [11]. In addition, when a user clicks on a plus or minus symbol, the center graph animates a grow or shrink operation, respectively. The animation provides prompt feedback and makes for a livelier user experience.

Discussion

CodeConnections scales very well. It can show the entire project at a glance, revealing trends in the code base. For example, during one investigation of *Azureus*, we noticed a particularly important file in the repository disappear. It was linked to over 50 other files by inheritance (upper left of Figure 2). Raw CVS data could never have called such attention to it. We imagine that *CodeConnections* will be of most use to developers, especially new developers, giving them insight into how all of the pieces fit together.

We hope that *CodeSaw*, on the other hand, will reveal trends in an author's life, similar to the experiences of users with *PostHistory* [5]. In Figure 4, for example, we imagine that the author depicted in red will remember precisely what caused the dramatic dips in productivity during the months of April and August. A vacation, perhaps? The author shown in cyan may comment on the large volume of mail sent between the end of February and the beginning of April, especially since the rest of the workload remains fairly constant.

In the coming months, we will conduct interviews with the *Azureus* authors to obtain qualitative feedback on *LifeSource*. After gathering this early data, we plan to conduct a more wide-ranging and formal study. We expect we will provoke a response from the ranks next to authors' names. It would certainly be interesting to know whether making this visualization available publicly (on *SourceForge.net*, for example) would have any impact on the productivity of the project.

Will authors work to raise their rank in *CodeSaw*? Will they blog about their effects on it? In this way, we see potential for *CodeSaw* as a social data analysis tool.

Acknowledgements

We thank the UIUC Social Spaces group. In particular, Tony Bergstrom and Mat Yapchaian offered valuable feedback during the design process.

References

- [1] Azureus
<http://azureus.sourceforge.net/>
- [2] Baker, M. J., Eick S. G. Space Filling Software Visualization. *Journal of Visual Languages and Computing* 6 (1995), 119-133.
- [3] Cyclomatic Complexity, *Wikipedia: The Free Encyclopedia* (Dec. 16, 2005)
http://en.wikipedia.org/wiki/Cyclometric_complexity
- [4] Froehlich, J. and Dourish, P. 2004. Unifying Artifacts and Activities in a Visual Tool for Distributed Software Development Teams. *Proceedings of the International Conference on Software Engineering ICSE* (2004), 387-396.
- [5] Nguyen, D. H., Potter, J., Viegas, F. B., Boyd, D., and Donath, J. Digital artifacts for remembering and

storytelling: PostHistory and social network fragments. *Proceedings of HICSS* (2004), volume 4.

[6] Orgell, E. More Than 1.1 Million Developers In North America Now Working On Open Source Projects.
<http://evansdata.com/n2/pr/releases/DPS2004.shtml>

[7] Processing
<http://processing.org>

[8] SourceForge.net.
<http://sourceforge.net>

[9] Viégas, F. B., Wattenberg, M., and Dave, K. Studying cooperation and conflict between authors with *history flow* visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2004), 575-582.

[10] Viegas, F. B., and Smith, M. Newsgroup crowds and authorlines: Visualizing the activity of individuals in conversational cyberspaces. *Proceedings of HICSS* (2004), volume 4.

[11] Wattenberg, M. Baby Names, visualization and social data analysis. *Proceedings of INFOVIS* (2005), 1 – 7.

[12] yFiles, yWorks
http://www.yworks.com/en/products_yfiles_about.htm