Visualizing the evolution of software using softChange

Daniel M. German, Abram Hindle and Norman Jordan

Software Engineering Group

Department of Computer Science

University of Victoria

{dmgerman,abez,njordan}@uvic.ca

June 22, 2004

This Presentation

- What am I going to cover?
  - Source Code Repositories
  - CVS
  - MRs
  - Questions that are applicable to source code repositories.
  - Previous Work
  - softChange
  - Summary

Source Code Repositories

- Products
  - CVS
  - Subversion
  - Clear Case
  - Source Safe
  - BitKeeper
- Functionality
  - Revisions
  - Branches
  - Concurrency
  - Configuration Management

CVS

- Why CVS?
  - Defacto Standard for Open Source projects.
  - Many mature Open Source Projects have open repositories to study.
  - Learn about Open Source Software Development processes.

Operations

- CVS Operations
  - commit
  - update
  - checkout
- We attempt to track CVS Commits by grouping revisions.

MRs

- What is an MR?
  - Modification Request
  - Programmer submits a modification of the source code to the repository.
  - For CVS - when a programmer commits changes.

Questions

- What questions do developers have? [Wu03],
  - What happened since I last worked on this project?
  - Who made this happen?
  - When did the change take place?
  - Where did the change happen?
  - Why were these changes made?
  - How have the files changed?
  - What methods or functions were changed?
  - What is the frequency of change?
  - Which files have changed?
  - Who is working on each module?

Questions

- What questions do administrators have?
  - How often does a programmer complete a MR?
  - How much does the programmer change per MR
  - What kind of commits does one programmer do?
  - How much changed between each release?
  - How many bugs are fixed and found after a stable release?
  - What kind of modifications are done at a certain time?
  - When was a module stabilized?
  - What is the daily LOC count for each programmer?
  - When is a module actively being developed and maintained?

Software Evolution

- Why study software in this manner?
  - Programmers are not always available for interview.
  - Provide historical evidence about software.
  - Correlate Project History to the Source Code.
  - Verify assertions about the project’s development.

Previous Work

- Previous Work
  - Xia is a plugin for Eclipse for the visualization of CVS repositories [Wu03]
  - Lrx [GG04] and Bonsai [Her04] provide Web Interfaces to the CVS Repository.
  - Fisher and Gall created a CVS fact extractor [FPG03]
  - Hippikat , by Davor Cubranic and Gail C. Murphy [CM03], combines many sources of data and provides queryable interface to search through this historical data.

softChange

- What is softChange ?
  - softChange is a collection of applications that work together in order to further study the software evolution of a project.
  - softChange elaborates on data provided from many sources to enable an accurate description of the evolution of a project.
  - softChange helps answer common questions maintainers, developers and administrators have about a project.

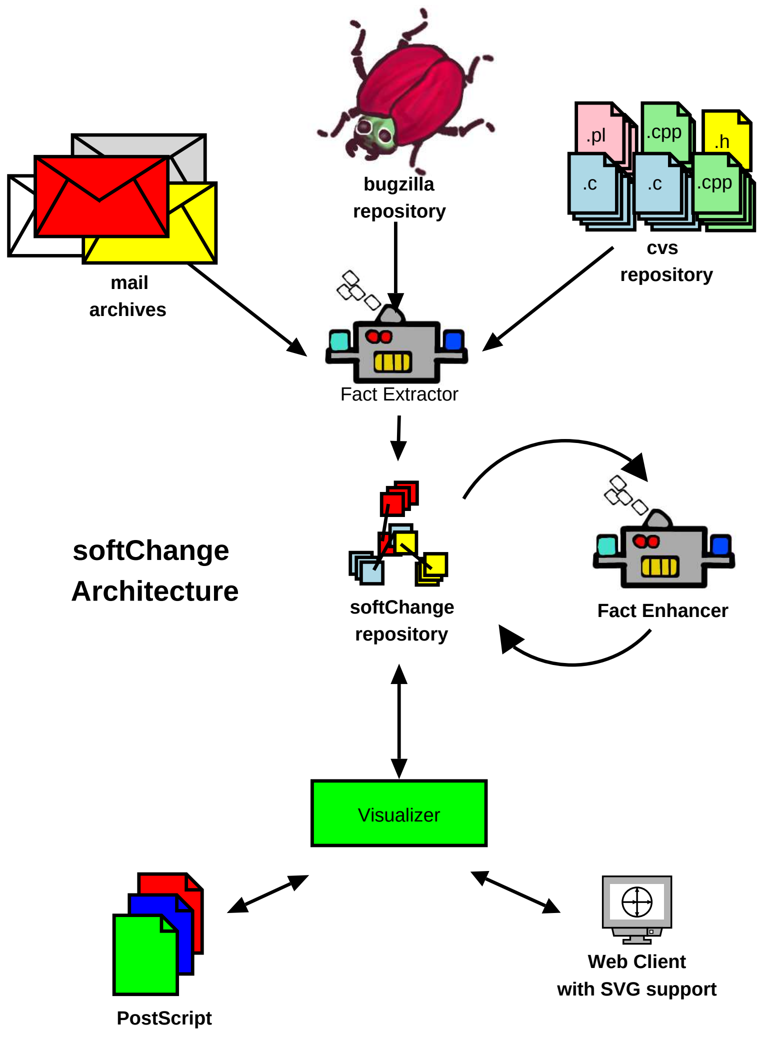


Figure 1: Architecture of Softchange

softChange

- What is softChange ?
  - Software Trails Repository - A relational database that stores all the historical data.
  - Software Trails Extractor - Extracts data from CVS, Changelogs, bug reports and emails.
  - Software Trails Analyzer / Fact Enhancer - Combines data in the repository to form MRs, and produce other useful statistics.
  - Visualizer - Visualize the data in the repository to aid the user in exploration and discovery.

Visualization

- What can Softchange Plot?
  - Growth of LOCS vs time, at the project level and at the module level
  - Number of MRs vs time: How many MRs are committed in a given period?
  - Number of files vs time: How many files are part of the project at a given point in time?
  - Number of files in a given MR
  - Proportion of MRs per contributor
  - Proportion of revisions per source code file: How frequently is a given file modified?
  - Number of modules that are modified in a given MR: How frequently an MR includes modifications of 2 or more modules?
  - Project time-tree: “When are given files created and modified?”, displayed in a timeline fashion.

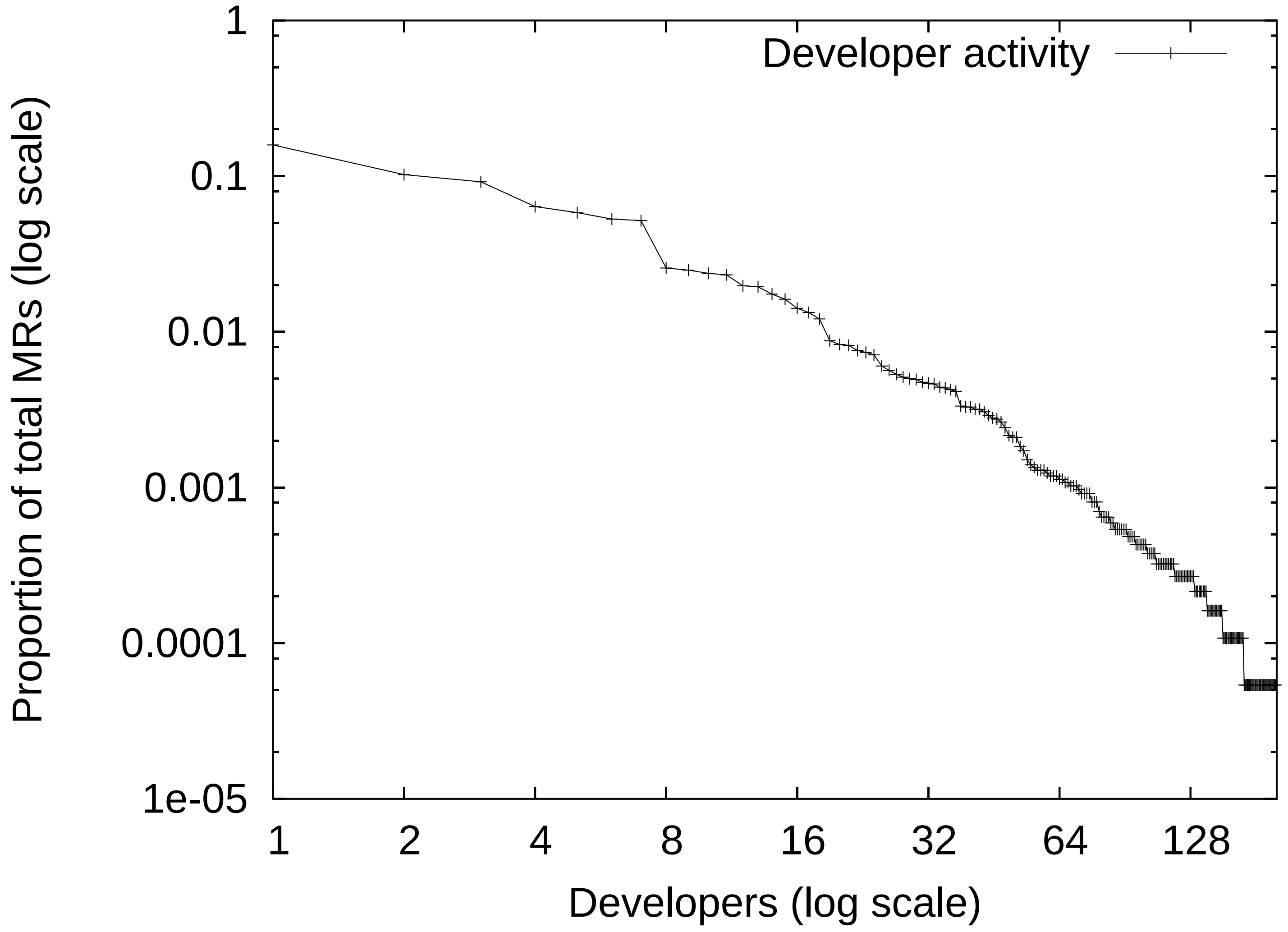
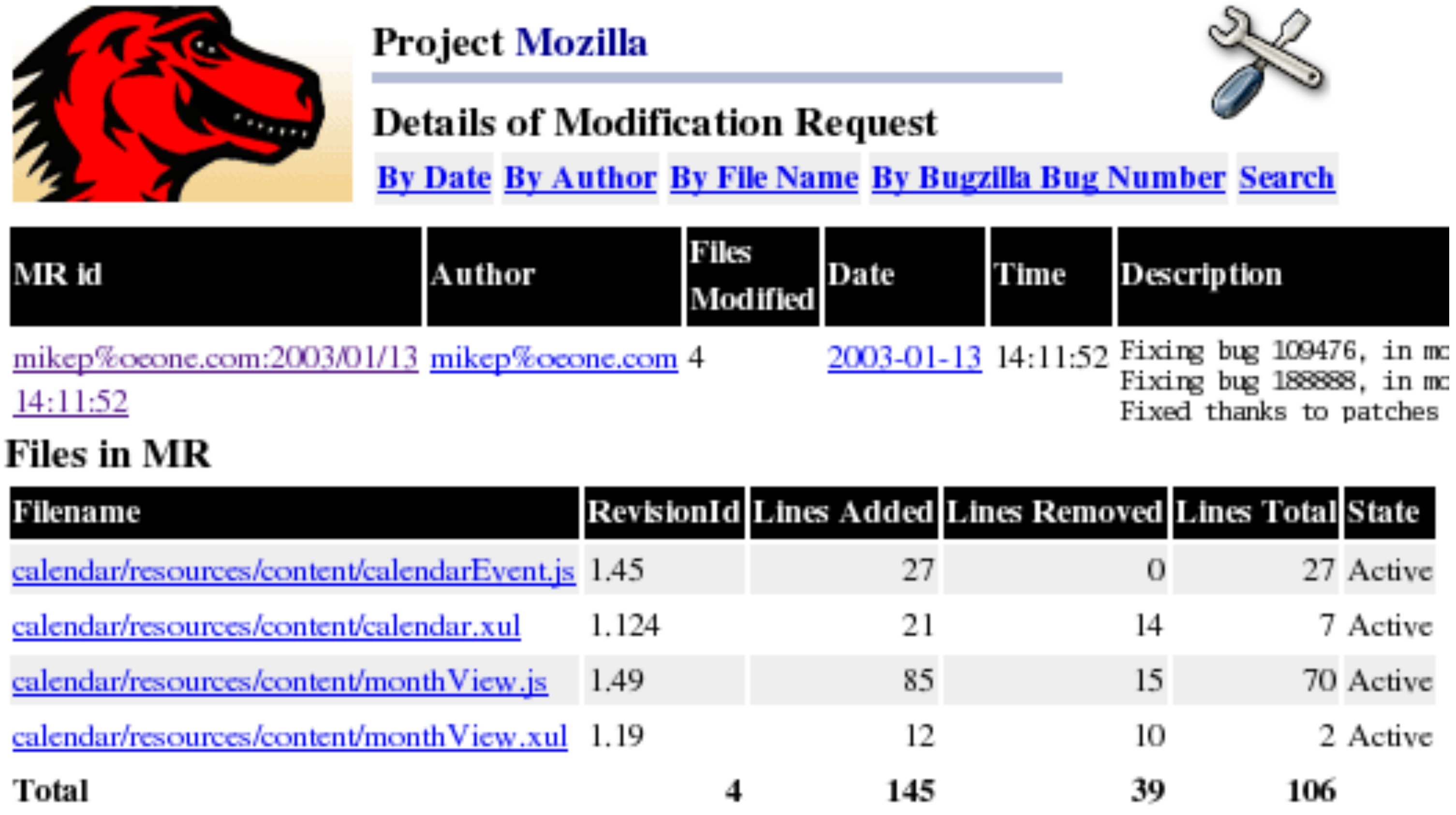


Figure 2: PostScript visualizer: proportion of MRs per contributor.



MR id	Author	Files Modified	Date	Time	Description
mikep%oeone.com:2003/01/13 14:11:52	mikep%oeone.com	4	2003-01-13	14:11:52	Fixing bug 109476, in mc Fixing bug 188888, in mc Fixed thanks to patches

Files in MR

Filename	RevisionId	Lines Added	Lines Removed	Lines Total	State
calendar/resources/content/calendarEvent.js	1.45	27	0	27	Active
calendar/resources/content/calendar.xul	1.124	21	14	7	Active
calendar/resources/content/monthView.js	1.49	85	15	70	Active
calendar/resources/content/monthView.xul	1.19	12	10	2	Active
Total	**4**	**145**	**39**	**106**	

Figure 3: Hypertext browser: details of an MR using softChange

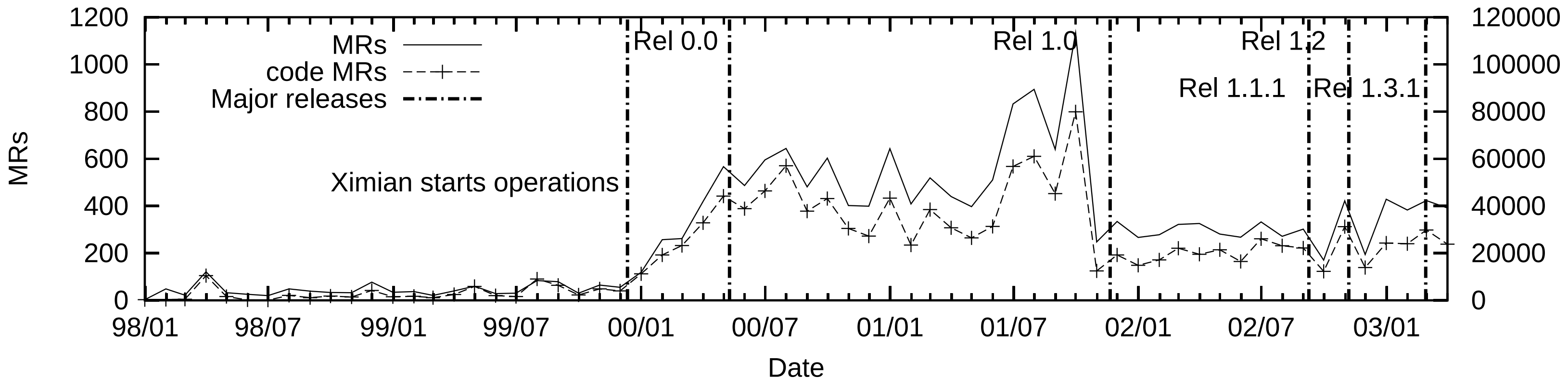


Figure 4: PostScript front-end: MRs over time.

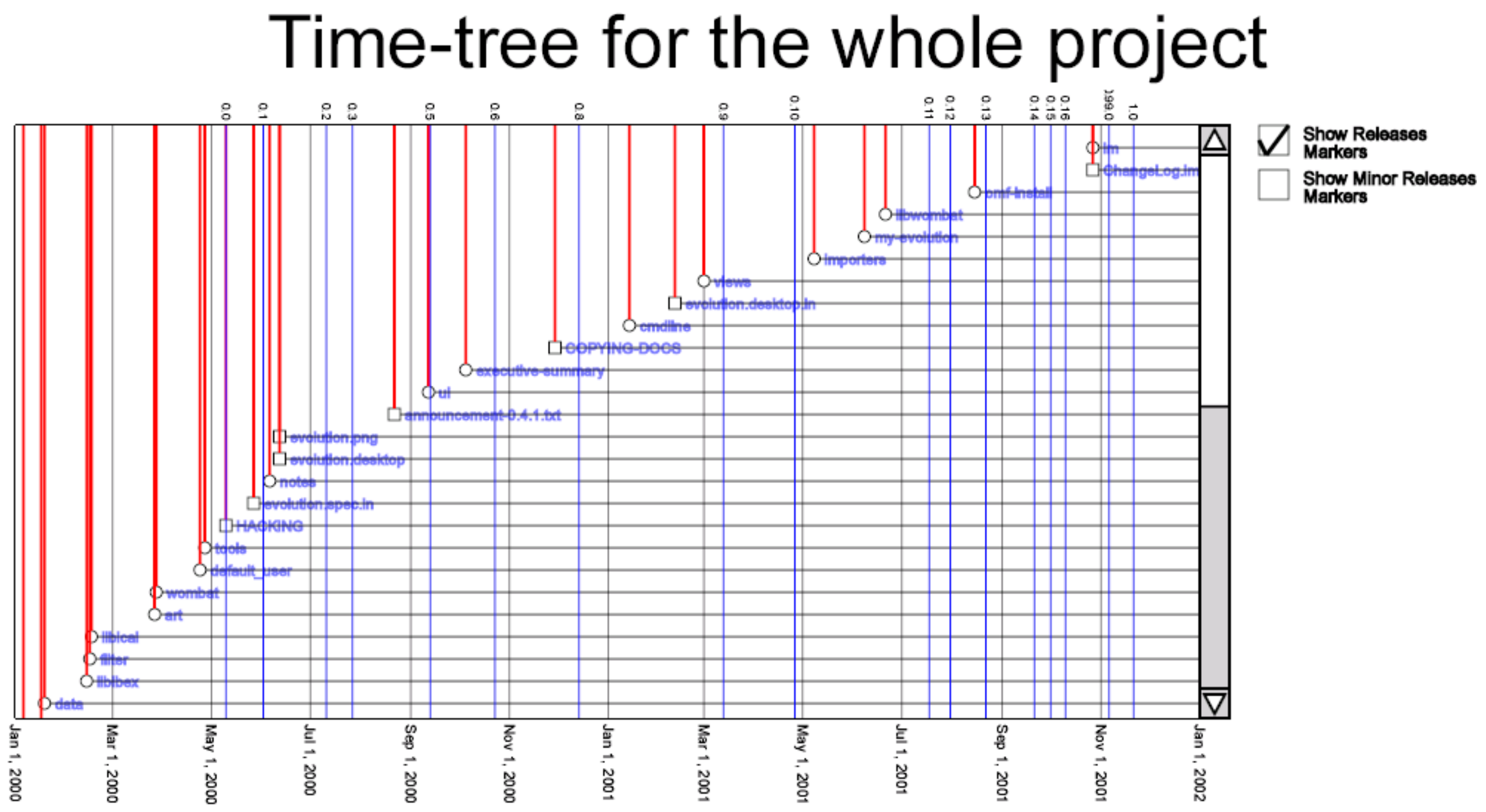


Figure 5: Time-tree in softChange

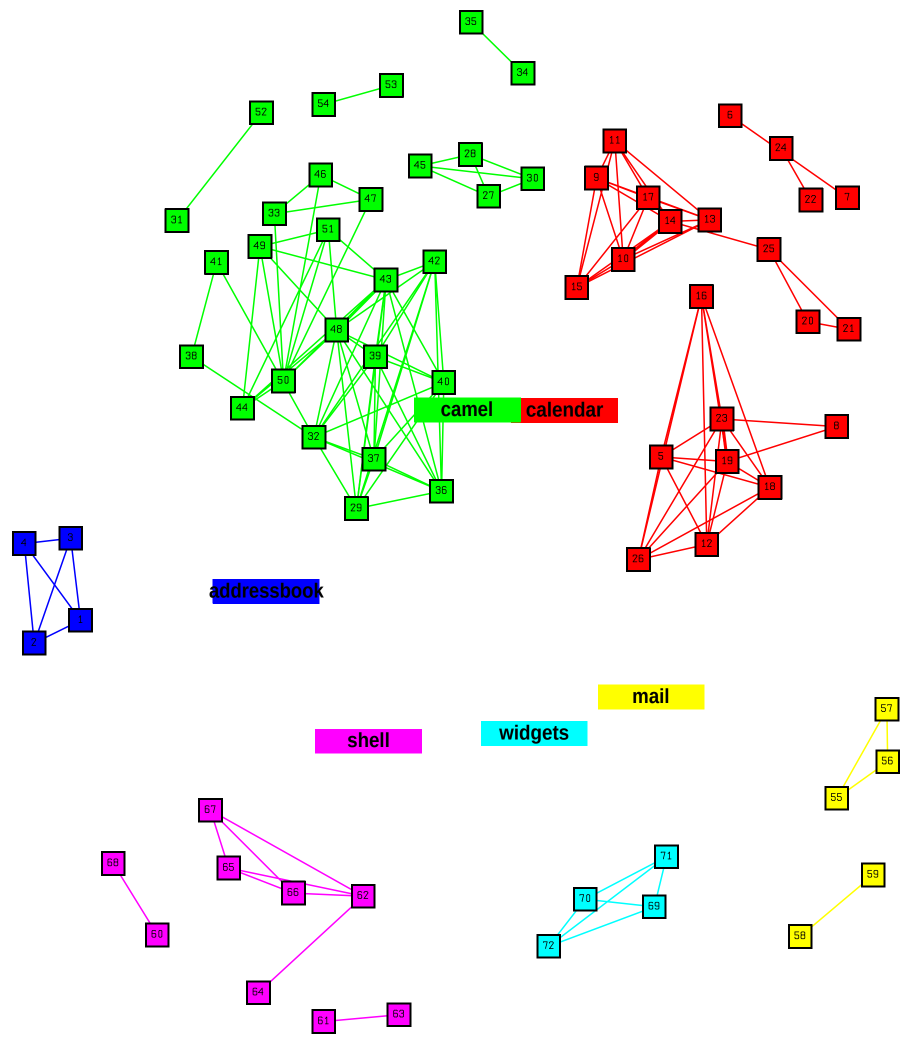


Figure 6: Evolution Modules 2002 10

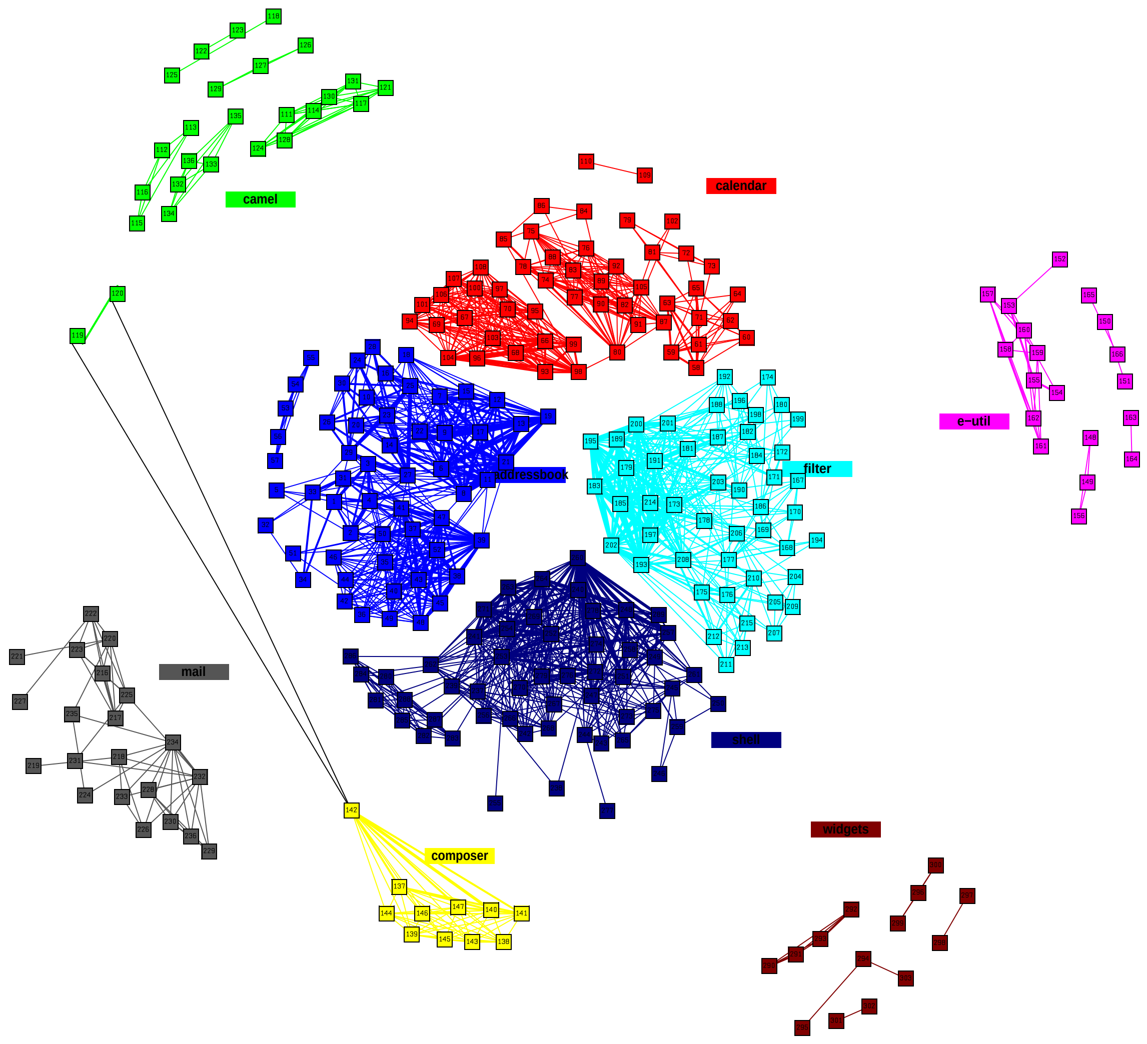


Figure 7: Evolution Modules 2002 11

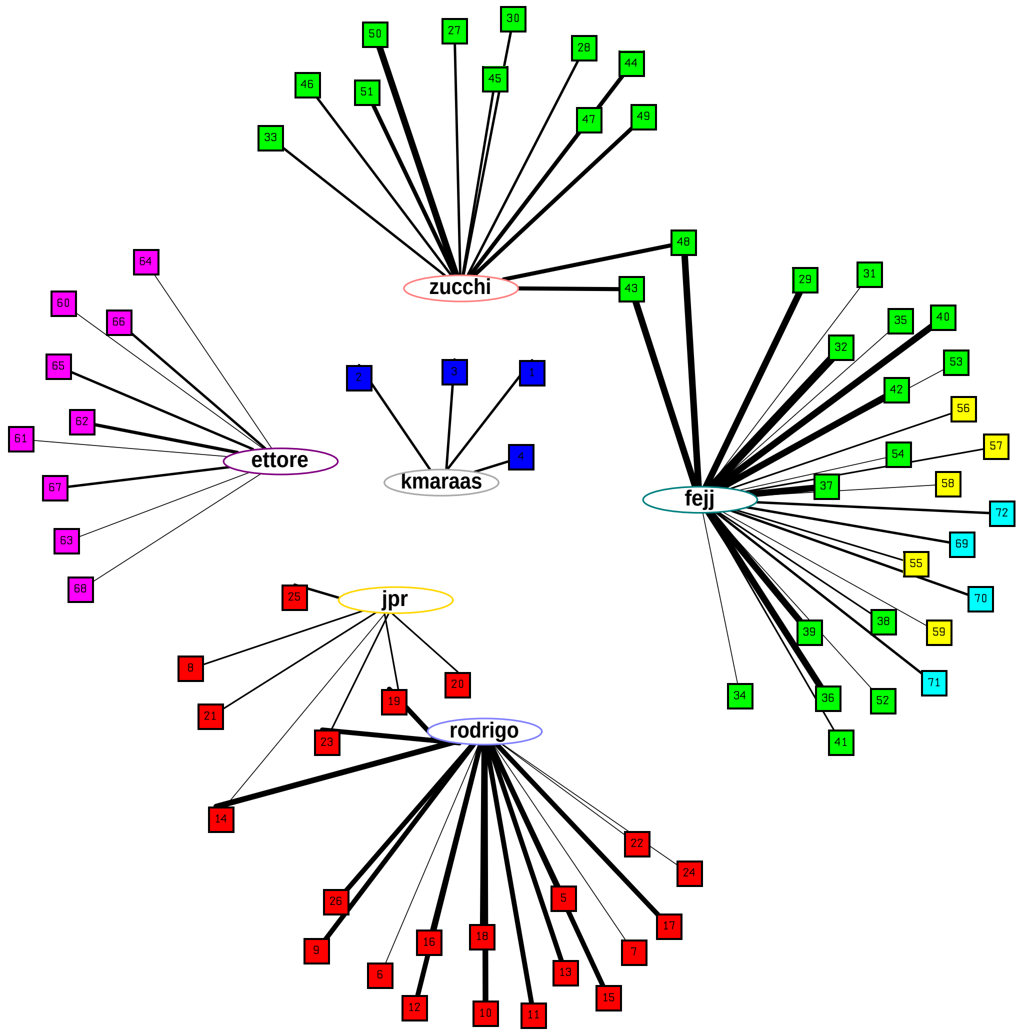


Figure 8: Author “Friendship” 2002 10

Visualizing the evolution of software using softChange

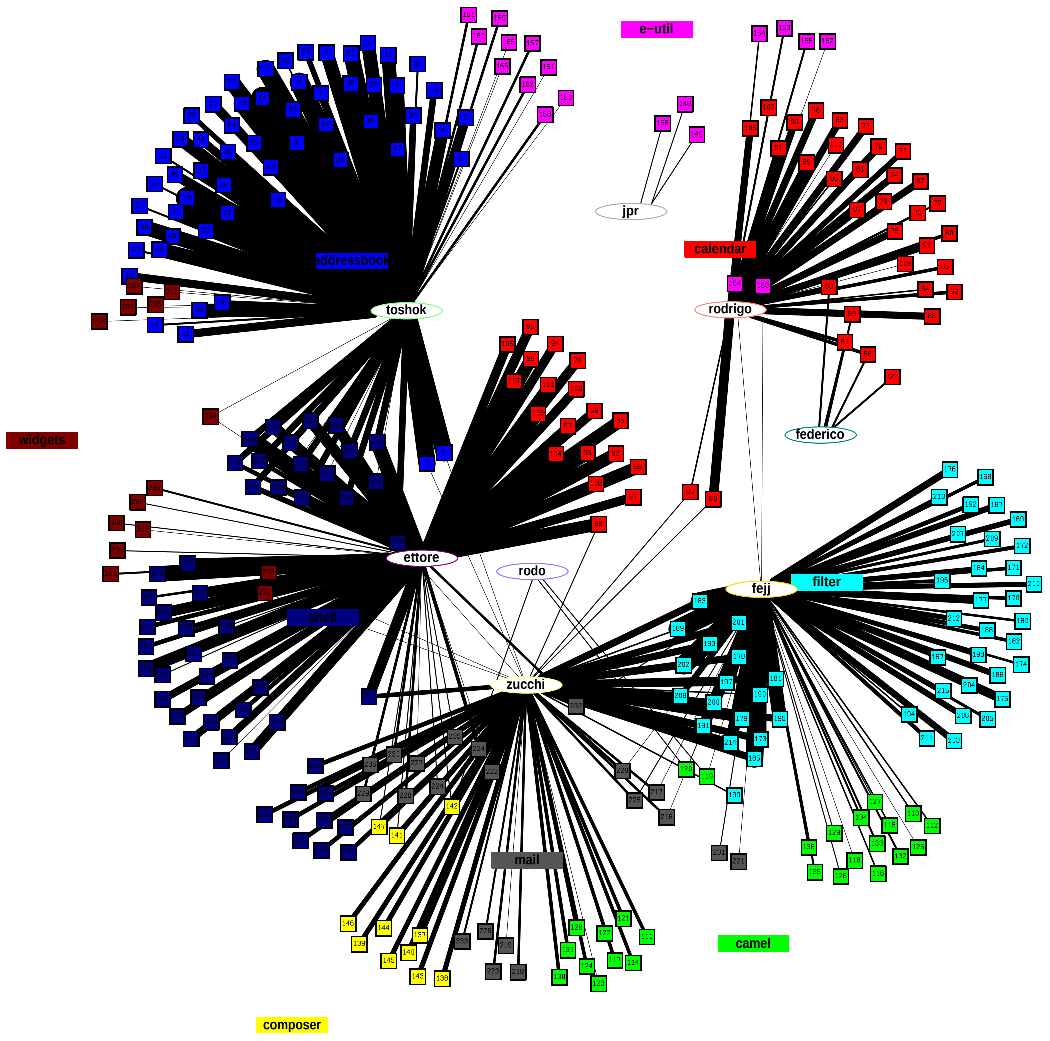


Figure 9: Authors “Friendship” 2002 11

Summary

- We have used softChange to understand the evolution of such products as Evolution and Mozilla. [Ger04b]
- We have used softChange to help describe how programmers collaborate on the GNOME project [Ger04a].
- The repository is extendable thus data maybe elaborated on without affecting other programs.
- Future Work involves further visualization of data in the repositories, classification of changes, and integration of softChange with other projects such as JReflex and Shrimp [SBM01].

References

[CM03] Davor Cubranic and Gail C. Murphy. Hipikat: Recommending pertinent software development artifacts. In *Proceedings of the 2003 International Conference on Software Engineering*, pages 408–418, Portland, May 2003. Association for Computing Machinery.

[FPG03] Michael Fischer, Martin Pinzger, and Harald Gall. Analyzing and relating bug report data for feature tracking. In *Proc. 10th Working Conference on Reverse Engineering*, pages 90–101. IEEE Press, November 2003.

[Ger04a] D. M. German. Decentralized open source global software development, the gnome experience. *Journal of Software Process: Improvement and Practice*, Accepted for publication, 2004.

[Ger04b] D. M. German. Using software trails to rebuild the evolution of software. *Journal of Software Maintenance and Evolution: Research and Practice*, To appear, 2004.

[GG04] Arne Georg Gleditsch and Per Kristian Gjermshus. lrx Cross-Referencing Linux. http://lxr.sourceforge.net/, Visited Feb. 2004.

[Her04] Tara Hernandez. The Bonsai Project. http://www.mozilla.org/projects/bonsai/, Visited Feb. 2004.

[SBM01] M.-A. D. Storey, C. Best, and J. Michaud. SHriMP Views: An Interactive and Customizable Environment for Software Exploration. In *Proc. of International Workshop on Program Comprehension*, May 2001.

[Wu03] Xiaomin Wu. Visualization of version control information. Master's thesis, University of Victoria, 2003.