

Visualizing commit comments with chromograms

Maike Maas Valerie Poser Sebastian Becking
Marc Jose Fabian Kosmale Heiko Becker

December 13, 2013

1 Project description

The project's main goal is to visualize code repositories with chromograms¹. We aim at supporting Git, Mercurial and Subversion with the main focus on git. The chromograms' purpose is to visualize the content of the repositories and not their structure as there already exists a wide range of tools supporting this usecase.

We hope that this approach will result in additional insight, especially for large projects.

2 Must-Haves:

1. The product's front end will be a website that will work with recent and well known browsers (Firefox 25.0, Internet Explorer 11, Google Chrome 31.0, Safari 7.0 and Opera 17.0).
2. The website will contain a home page, with an input field, a visualize button, a help button and a documentation button.
3. The input field is used to enter a URL to a repository.

¹“A chromogram is a visualization method that converts textual data to colours, producing a data-dense display that can fit a vast edit history onto a single image.”[1]

4. The visualize button starts visualizing the provided repository and opens the rendering page.
5. The help button displays an overlaying help box.
6. The documentation button opens the documentation page.
7. The home page will also have three buttons to automatically insert an example URL into the input field [example repositories are “JQuery”, “Twitter Bootstrap” and “Git”].
8. The documentation page will provide an detailed description of the project and a button that directs the user back to the home page.
9. The system will support visualization of the commit comments of a provided git repository by representing every commit comment as a colored block.
10. A repository will be provided by the user by entering a URL which refers to the repository that should be visualized.
11. The rendering page will display a progress bar while the system is rendering the picture.
12. The rendering page will show the visualization after the rendering process is finished.
13. The visualization will be displayed in a single picture.
14. The visualization will provide an overall block view of all commit comments. Overall means that one image will be displayed for all commits, not segmented by commit time or modified files. Block view means that the image is a rectangle consisting of multiple rectangles representing each commit message.
15. The main visualization parameter will be the commit message.
16. The user will be able to select the number of commit messages to visualize.
17. The product will be released open source.
18. The application language will be English.

19. The application will detect if the repository's hosting server doesn't answer and will inform the user about that fact by printing an error message.
20. The application will detect if it has no access to the remote repository and will inform the user about that fact by printing an error message.
21. The system will support repositories whose commit comments are encoded in ANSI-ASCII. It will skip over Unicode glyphs.
22. The application will run on a Linux server having PHP 5.3 and Apache/2.2.24(Unix) installed.

3 May-Haves:

1. The product may support visualizing commit comments of mercurial and subversion repositories analogously to how git repositories are visualized.
2. There may be an extra customize page where the user may be able to customize the visualization.
3. The user may be able to customize the color-mapping for the parameters on the customize page.
4. The user may be able to select which parameters he wants to influence the visualization on the customize page. In addition to the visualization parameters mentioned before the system may support parameters from the list below (see May-Have 5).
5. The application may support these parameters for the visualization:
 - a) user
(the person that made the commit to the repository)
 - b) kind of change
(the numbers of lines that were added, modified or deleted with the commit we're working with)
 - c) only added something
(taking only the commits into account that added lines or files)

- d) only deleted something
(taking only the commits into account that removed lines or files)
 - e) time of day of commit
(the time of day at which the commit we're currently working with was made)
 - f) date of commit
(the date at which the commit we're currently working with was made)
 - g) structure of the repository
(branches, merges)
6. The parameters may influence a block's color, size, alignment and order.
 7. The visualization may be supported by a legend mapping used colors to the corresponding words.
 8. The product website may be responsive and may also work with mobile clients. In this context responsive means, the layout will be optimized for the current browser used by the user.
 9. The visualization may be downloadable as a PDF or JPEG.
 10. The website may show the URL from a random repository of Github as example.
 11. The user may click on a single block within the visualization to see further information about the specific commit comment represented by that block.
 12. The application may be available in multiple languages.
 13. The visualization of the repository created on the rendering page may be scalable.
 14. The application may cache repositories that have already been displayed by the user.

4 Won't-have:

1. The product will not have a desktop client and cannot be used without a web browser.
2. The client will not support version control systems other than Mercurial, Subversion and Git.

3. The software will not have a native app for any mobile operating system.
4. The user will not be able to upload a folder of a git repository instead of an URL.
5. The application will not provide a user login interface where users can manage their pictures or their repositories.
6. The visualizations will not be shared on Google+, Facebook, Twitter and Instagram.
7. The application will not provide access to the repositories.
8. The application will not provide access to protected repositories.
9. The application will not create animated renderings.

References

- [1] Martin Wattenberg, Fernanda B. Viegas, Katherine Hollenbach, *Visualizing Activity on Wikipedia with chromograms*. 2007.