Analyzing Collaborative Artefacts in Project Based Courses

Vassiliy Platonovitch Tchoumatchenko

Abstract – **The paper explores the problems and suggests some solutions in evaluating contributions of each author in collaboratively created artefacts. It reflects our experience of using Google Docs and other cloud based instruments in project based courses for undergraduate students. A new tool for analyzing document's revisions and contributions is discussed.**

Keywords – **Google Docs, collaborative writing, project based learning.**

I. Introduction

In the last two years, several engineering courses at the Technical university of Sofia were re-designed using trialogical design principles and modern digital technology. The courses were restructured from traditional face-to-face to project oriented adopting and applying modern online learning platforms, cloud collaboration tools and social software.

Introducing new technologies and paradigms in established engineering courses is always challenging. In addition to the core subject matter, students had to learn new tools and development workflows.

Overall, the trialogical approach was well accepted and considered as an appropriate path for transforming students' individual course work into more collaborative activities.

Writing collaboratively, however, takes coordination and awareness of who has done what. Each student's activity and contributions to the collaborative project is influential (but not definitive) in determining the final grade. On the other hand, being able to analyze how the project report evolved over time can reveal interesting patterns of collaborative writing.

II. Related Work

Collaborative writing is on the increase and many researchers have created tools to analyze documents evolution. One such tool, DocuViz [1], displays the entire revision history of Google Docs, showing more than the one-step-at-a-time view now shown in revision history. DocuViz is potentially useful in cases such as: To authors themselves to see recent "seismic activity," indicating where in particular a co-author might want to pay attention, to instructors to see who has contributed what and which changes were made to comments from them, and to researchers interested in the new patterns of collaboration made possible by simultaneous editing capabilities.

V. Tchoumatchenko is with the Department of Electronics and Electronics Technologies, Faculty of Electronic Engineering and Technologies, Technical University - Sofia, 8 Kliment Ohridski Blvd., 1000 Sofia, Bulgaria, e-mail: vpt@tu-sofia.bg

Another tool for analyzing Google Docs history is Draftback [2]. It shows the timeline of the changes, and below it, a "map" that tells where in the document each of those revisions happened. Draftback is implemented as a Chrome extension and is able to playback the complete history of every single character.

Unfortunately, none of these tools is open source. This makes it difficult to adapt and integrate them in our collaborative learning infrastructure.

III. Infrastructure for Collaborative Learning

The infrastructure for collaborative learning [3] consists of public cloud based services, combined in a way that supports electronic design workflow (fig. 1). Working in small teams, the students are required to design a digital integrated circuit. The design workflow is based on HDL modelling, verification and synthesis. The main design artefacts (VHDL models and test-benches) are text files; therefore we are able to borrow many tools and workflows from the software development community. Projects are hosted on GitHub [4] – one repository per project. In parallel with the code development, the teams are required to create and maintain a Google Docs document which is one of the major deliverables. Initially the document contains the technical specifications of the design. Later on, the students have to add description of the implemented algorithms and architectures, argumentation of the tradeoffs made and the results from the simulation, synthesis and physical design. Most of development takes place outside the regular classes. For their intra-team communication, the students are free to choose whatever tools they prefer (chat, conferencing, email). For student - teacher communications we decide to use the Google tools: Gmail, Docs, Talk, Calendar, Drive and Google+. Students were encouraged to submit their questions as emails instead of chat messages.

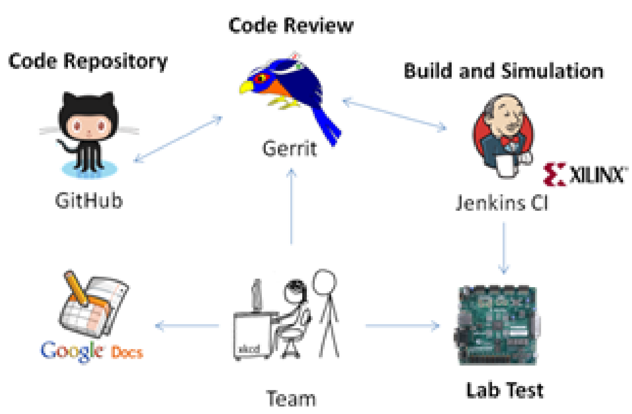

Fig. 1. Collaborative workspace

IV. Collaborative Artefacts Evolution

A. GitHub Revision History

GitHub is optimized for hosting software projects. It provides a very detailed history of commits for each repository (fig. 2). Each commit is attributed to an author. A single commit usually consists of changes in multiple files. Each change can be individually inspected (fig. 3). In the majority of cases, the tools provided by GitHub are more than adequate for analyzing the evolution of the students projects.

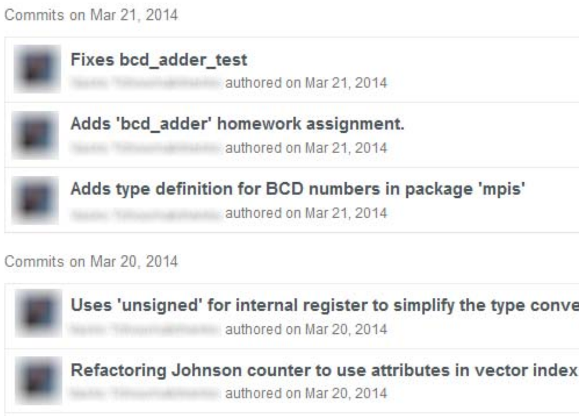

Fig. 2. GitHub commits history view

Fig. 3. GitHub diff view

B. Google Docs Revision History

The functionality offered by Google Docs with respect to exploring documents history is rather limited. At a file level, there is an activity view (fig. 4), that provides a good overview of when and who created or modified a particular document.

At document level, we have a revision history (fig. 5) which shows a timeline of the changes, but no information about the scope of each change. Therefore a simple formatting modification and a substantial text contribution are indistinguishable in the revision history view. Clicking on a particular revision, reveals the document content with all relevant text changes colored. It's quite frustrating that there is no way to quickly locate the changes – the user has to scroll through the document and look for a colored text. Some changes as added or deleted figures are not indicated at all.

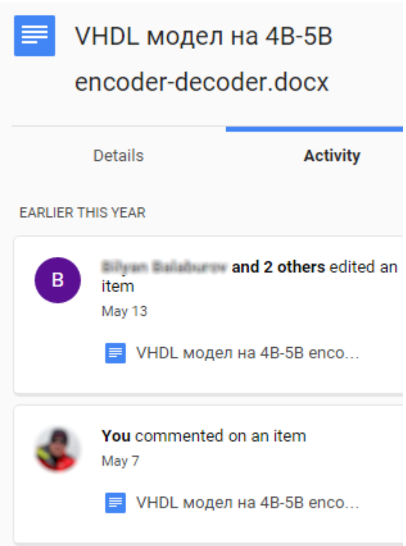

Fig. 4. Activity view.

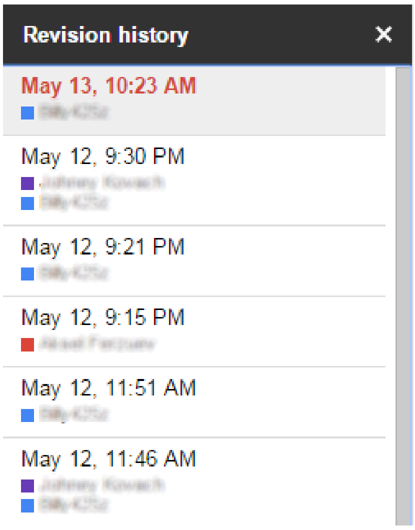

Fig. 5. Revision History View

V. A New Tool for Exploring Google Docs History

To facilitate the exploration of collaboratively created project artefacts, we developed a new application for analyzing Google Docs revision histories. The following design requirements were specified:

- The changes in each revision should be visualized in a way, similar to the one used by GitHub (fig. 3).
- The application should find word-level changes.
- Formatting changes (e.g. fonts, colors) should be ignored.
- It should be possible to show all contributions of a particular author.

```
8B/10B encoder/decoder е устройство което, [-първо кодира 8 битова-]{+извършва както кодиране на 8-битова+} дума в [-10 битова, след ^M-]
[-което информацията се пренася-]{+10-битова, ^M+}
{+така+} и [-се предава на декодиращо устройство, което декодира думата ^M-]
[-от 10 бита отново на 8.-]{+обратното декодиране.+} На схемата долу е представена точно такава система. Тя се използва
за високоскоростно, серийно предаване на информация. Енкодера на страната на трансмитера е
съставен от 8 битов паралелен вход и 10 битов изход. Този 10 битов изход е зареден във
високоскоростен [-Serializer (това е 10 битов Shift регистър с-]{+преобразувател от+} паралелен [-вход и сериен ^M-]
[-изход).-]{+в последователен код .+}
След [-което-]{+това+} информацията се предава до високоскоростен Deserializer (10 битов Shift
регистър с паралелен изход и сериен вход) в страната на приемника и се преобразува от
серийна до паралелна. Декодера [-обръща информацията-]{+преобразува данните+} от 10 [-битова до-]{+битови в+} 8 [-битова.-]{+битови.+} Когато се
използва 8B/10B кодираща схема, серийното предаване на информацията е DC – балансирано,
това означава че се изпраща еднакво количество от нули и единици за дадената дължина на
предаваната информация и максимална run-length без преходи от 5. Run-length се дефинира
като максималния брой на нулите и единиците предавани в серийния пренос на информацията.
Тези две характеристики помагат във възстановяването на информацията и [-clock-a-]{+синхронизация на^M+}
{+тактовия сигнал+} в приемника.
```

Fig. 6. Differences between two revisions – wdiff format

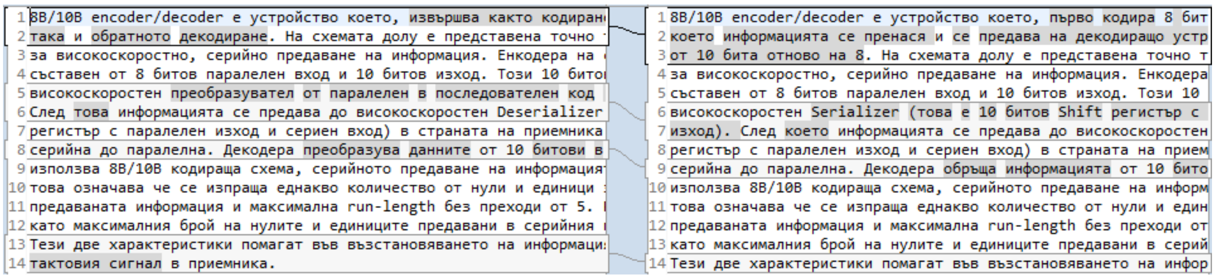

Fig. 7. Differences between two revisions – UI mockup

- The application should use public Google Docs API [5].
- The application should be cross-platform – both desktop and mobile devices should be supported.

To fulfill the cross-platform requirement, the document history exploration tool was implemented as Google Chrome extension [6]. This allows for a natural UI integration – the user can open a Google Docs document in her Chrome browser and then start the application from the browser's toolbar.

The application's UI is still work in progress. Presently, the differences between revisions are shown as text based output (fig. 6). The added and deleted words are marked in a way similar to the output of the wdiff utility [7]. In the final implementation, the compared text will be shown next to each other and the differences will be indicated by different colors (fig. 7).

VI. Conclusions

We presented in this paper our experience in analyzing the artefacts of collaborative design projects. We have implemented an application that shows the changes in each document revision and the contributions of each author in a more usable format than the native Goggle Docs revision history. We believe that such tool can be useful both for the authors of the document and for the professor, who evaluates the project.

Acknowledgement

The work in this paper is a part of the EU project "Promoting Knowledge Work Practices in Education – KNORK" at the Department of Electronics - Technical University of Sofia, which is supported by the Lifelong Learning Program of the European Community.

References

[1] D. Wang et al. DocuViz: Visualizing Collaborative Writing. Proc. of the 33rd Annual ACM Conference on Human Factors in Computing Systems, ACM, pp. 1865-1874, 2015.
[2] J. Somers, 'How I reverse-engineered Google Docs to play back any document's keystrokes', 2015. [Online]. Available: http://features.jsomers.net/how-i-reverse-engineered-google-docs/. [Accessed: 05- Jul- 2015].
[3] T.Vasileva, V. Tchoumatchenko, S. Stefanova, Introducing Trialogical Knowledge Practices in Higher and Secondary Education, The 3rd Electronic International Interdisciplinary Conference (EIIC), Virtual Conference, September, 1 - 5, Slovakia, 2014, pp. 580-585, ISBN: 978-80-554-0921-4
[4] https://github.com/
[5] https://developers.google.com/drive/v2/reference/
[6] https://developer.chrome.com/extensions
[7] http://www.gnu.org/software/wdiff/