WriteProc: A Framework for Exploring Collaborative Writing Processes

Vilaythong Southavilay, Kalina Yacef

School of Information Technologies, The University of Sydney
NSW 2006, Australia

vstoto@it.usyd.edu.au, kalina@it.usyd.edu.au

Rafael A. Calvo

School of Electrical and Information Engineering, The University of Sydney
NSW 2006, Australia

rafa@ee.usyd.edu.au

Abstract *Collaboration and particularly collaborative writing is an increasingly essential skill needed in the workplace and education. Until recently most of the focus of research has been the final product of the writing, rather than the process itself. In this paper, we propose an innovative framework for investigating collaborative writing processes. The WriteProc framework utilizes both process and text mining tools to analyze the process that groups (or individual) writers follow, and how the process correlates to the quality and semantic features of the final product. Furthermore, WriteProc is integrated with existing web 2.0 writing tools, providing full support for writing, reviewing and collaboration. We describe the architecture that integrates tools for analyzing the process and semantics of the writing. We also provide a case study on data collected from a group of undergraduate students writing collaboratively an essay, with peer reviewing and use of an automatic feedback tool.*

Keywords Document workflows, web documents, process mining

1 Introduction

Computer-Supported Collaborative Work (CSCW), particularly Collaborative Writing (CW), has received attention since computers have been used for word processing. Due to the availability of the Internet, people increasingly write collaboratively by sharing their documents in a number of ways. Writing individually and collaboratively are considered essential skills in most industries, academia, and government. This has led to increased research on how to support the production of better documents.

In Education, computer-supported writing has been studied for decades. Goldberg et al. [6] collected a decade of empirical data and in a meta-study found "that when students write on computers, writing becomes a more social process in which students share their works with each other". They also noted that when using computers, students prefer to make revisions while producing, rather than after producing, text. Between initial and final drafts, students also tend to make more revisions when they write with computers. In most cases, students also tend to produce longer passages when writing with computers. In addition, review feedback, especially peer review, has been recognized as one effective way to learn writing [3, 4]. When students write with computers, they engage in the revising of their work throughout the writing process, more frequently share and receive feedback from their peers, and benefit from teacher input earlier in the writing process. Although these studies show that computer-supported writing including automatic feedback tools efficiently assists students in writing and reviewing, understanding the writing process is crucial for developing support technologies for CW.

Over the past two decades, there has been abundant text-mining research for improving the support of quality writing. But work such as automatic scoring of essays [11], visualization [9], and document clustering [1] focus on the final product, not on the writing process itself. Our vision is to investigate how ideas and concepts are developed during the process of writing could be used to improve not only the quality of the documents but more importantly the writing skills of those involved.

Improving the process of writing requires understanding how certain sequence patterns (i.e. the steps a group of writers follow) lead to quality outcomes. We see the sequence pattern as comprised both of time events (as used in other process mining research) and of the semantics of the changes made during that step.

We combine here two techniques: process mining, which focuses on extracting process-related knowledge from event logs recorded by an information system, and semantic analysis, which focuses on extracting knowledge about what the student wrote (or edited). The

Proceedings of the 14th Australasian Document Computing Symposium, Sydney, Australia, 4 December 2009.

field of process mining covers many areas, like performance characteristics (e.g. throughput times), process discovery (discovery of the control flow), process conformance (checking if the event log conform specification), and social networks (e.g. cooperation) [2]. Particularly, process mining analysis is necessary to understand group awareness, and writers' participation and coordination. Text mining combines indexing, clustering, latent semantic analysis and other techniques studied by the document computing community.

In this paper, a conceptual framework and tools for supporting collaborative writing (CW) are introduced. Our framework is based on a taxonomy of collaborative writing proposed by Lowry et al. [8] and defines writing activities, strategies, work modes and roles involved in CW. With this taxonomy, the framework incorporates process mining and text mining technologies in order to gain insight of collaborative writing process.

The remainder of the paper is organized as follows. In Section 2, WriteProc, a framework for supporting CW and the analysis of its process and semantics is presented. A case study of process mining for a reviewing tool, Glosser is then presented in Section 3. Finally, Section 4 provides discussion of our case study and future work planned in this area.

2 WriteProc

Let us describe WriteProc, a framework for analyzing individual and collaborative writing process. It consists of three tools: writing, reviewing and analysis tools. The analysis tool utilizes both process and text mining techniques.

Our aim of developing WriteProc is to assist individual or groups of writers during the writing process. Particularly, WriteProc can advise writers with reviewing feedback and visualization of the analyses of writing activities and text changes during the process of writing.

2.1 Overall conceptual description

The framework integrates a front-end writing tool which not only supports collaborative writing activities, but also stores all revisions of documents created, shared and edited by groups of writers. Each revision of particular documents must contain all needed information such as edited text, timestamp of committing change, and identification of the writer. In order to perform analysis of writing process for particular documents, all revisions of the documents are retrieved and traced.

A reviewing tool is also embedded in the framework. It assists writers in revising their own pieces of writing and reviewing others works. After receiving feedback generated automatically by the reviewing tool, writers can edit and change their documents' content accordingly. The tool keeps records of writers' reviewing activities in event logs. The event logs of the tool are then extracted to gain an insight on how writers use the reviewing tool and how review feedback affects changes in reviewed documents.

Process and semantic analysis tools are used in the framework. Based on both the information (such as timestamp and writers' identification) of all revisions and event logs of reviewing activities, a process mining tool is used to discover sequence patterns of writing activities. The process analysis provides a way to extract knowledge about writers' interaction and cooperation. The analysis can identify interactions' patterns that lead to a positive outcome and indicate patterns that may lead to problems. In addition, a text mining technique is performed to analyze text-based changes of all revisions of documents. The text-based analyses can provide semantic meaning of changes in order to gain insight into how writers develop idea and concept during writing process.

2.2 Implementation

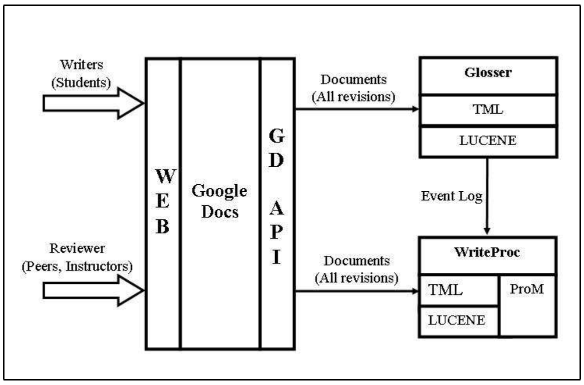

Figure 1: WriteProc: A framework supporting collaborative writing.

Based on the overall concept described above, the framework utilizes process and text mining technologies. It employs open-source utilities of those techniques to conduct analysis of writers' interaction and text in order to assist writers in identifying and realizing their writing process in collaborative manner. Figure 1 shows the framework for supporting collaborative writing (CW).

2.2.1 Writing environment: Google Docs

In order to use a process and semantic analysis tool in real scenarios, the tool must be closely integrated to the writing environment. Tools such as Microsoft Word or OpenOffice do not keep traces of the writing process. Web 2.0 tools such as Google Docs (and the incipient Microsoft Word Live) allow users to write on a web application (or offline and then synchronizing). The service provider keeps the different versions of the document. Therefore, we selected Google Docs in our implementation of WriteProc.

In WriteProc, Google Docs (GD) is used as a front-end writing tool of the CW. It is a web-based utility with most needed functionalities for word processing and it allows users to share their documents with other team members and to write synchronously. Users can access GD through their web browsers from anywhere and at anytime they want. Each user needs a Gmail account to access the tool that they can obtain from Google free of charge.

At the center of the framework is Google Document Lists Data API (GDAPI) used to integrate GD to our CW system as shown in Figure 1 The API allows WriteProc to retrieve and track all versions of documents created, shared and edited among groups members. In GD, each document created is uniquely assigned a document identification number. The GD also keeps track of all version numbers of each document by incrementing its version numbers each time the document is edited. Every time a writer makes changes and edits a particular document, the identification of the writer, the edited content of the document, timestamp of committing changes and the version number of the edited document can be retrieved and stored at the central relational database of CW system by using the API. This information extraction is executed seamlessly offline and users as writers are not aware of it and are able to perform their writing tasks seamlessly. The API also provides us the ability to build an interface to create and share documents in CW system. This can be very helpful for instructors or supervisors to create and assign documents to groups of writers and reviewers without accessing GD. An appointed owner of a document can edit it, where as an assigned 'viewer' can only review it.

2.2.2 Reviewing tool: Glosser

Glosser is a web-based application providing support for writing in English [16]. It was designed and implemented to support a review feedback model. Figure 2 shows such a model. Glosser assists users to revise their own document and review other documents. It has the analysis and revision tracking system used for reviewing. Writers can use Glosser in order to gain insight into their essays' structure and coherence. To review particular documents, the system consists of several functionalities, as shown in Table 1:

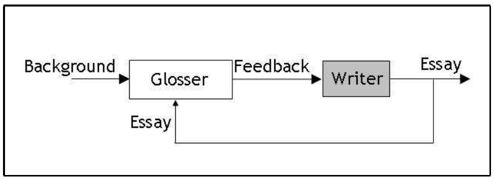

Figure 2: Automated writing feedback.

In the case study described in Section 3, students used a reviewing tool, Glosser [16]. A document created and shared among a group of writers can be reviewed in Glosser, which also accesses each revision using the Google Document Lists Data API. Users can access Google Docs from Glosser or vice versa.

Tool	Description
Home Tool (HOT)	showing basic statistics such as numbers of words and revisions.
Topic Tool (TOT)	checking if content provides evidence to support its topic sentences.
Flow Tool (FLT)	reviewing coherence and checking how paragraphs and sentences follow from previous ones.
Keyword Tool - HTML (KTH)	showing semantic flow.
Keyword Tool - Graph (KTG)	depicting the visualization of semantic flow.
Group Tool (GRT)	showing participation of authors for different versions.

Table 1: Reviewing tools of Glosser

2.2.3 Process and text mining tools

The interesting components of WriteProc are the process and text mining tools. The event log of Glosser is stored at the central relational database. The event log is used as a source to a process mining tool in order to gain an insight on writing activities and writers' interaction. The process mining tool utilized in the WriteProc is ProM [15]. In the next section, ProM will be used to demonstrate a process mining technique for our case study. In addition, an independent measure is developed to analyze the changes in each version of the documents in order to understand the nature of changes and the level of these changes. The analysis uses a text mining technique to find semantics changes among all versions of documents. This technique uses information from all the versions of documents performed by groups of writers. The text information of each version stored in the central database is indexed using Lucene [7] so that text produced by a group of writers can be systematically searched, sorted, filtered, and highlighted. After indexing all versions of documents, the system then analyzes the relationship between them and their terms using Text Mining Library (TML) in order to produce a set of concepts and nature of text changes in all versions of the documents.

3 Case study

As a way of evaluating the architecture and implementation of WriteProc and illustrating how it can be used, we discuss a case study where the tool is used to study writing processes in a software engineering unit conducted during the first semester of 2009 at the University of Sydney. It is important to note that Human Research Ethics Clearance has been completely granted from the university for this study. All students involving in the study signed an informed consent.

There were 58 students in the course, which was E-business Analysis and Design. They were organized in groups of two and asked to write Project Specification Documents (PSD) for their proposed e-business projects. Each group had to submit one PSD of between 1,500 and 2,000 words (equivalent to 4-5 pages). Students were required to write their PSD on Google Docs and share the documents with the course instructor. They were asked to submit their PSD using Glosser, a reviewing tool mentioned in Section 2.2.2. The submitted PSD was reviewed by other two students who were members of different groups. Students had one week to review each others' documents and submit their feedback. After getting feedback on their documents from their peers, students could revise and improve their writing if necessary before submitting the final version one week later. The submission of the final version of PSD also used Glosser. The total event log file of the system consisted of usage data of Google Docs and Glosser for three weeks. In addition to this log file, the marks of the final submissions of the PSD together with a very good understanding of the quality of each group through the semester was used to correlate behaviour patterns to quality outcomes. In particular, to be able to give insight into how students used the reviewing tool for revising their own documents and reviewing others and to give recommendation to improve the system, we performed a process diagnostics method to give a broad overview of students' interaction and collaboration.

3.1 Log Preparation

Unlike data preprocessing for workflow mining [5], our approach used a data preprocessing method for behavior pattern mining [12]. This method was used with a process mining tool like ProM [15]. Glosser's event log was a typical Web server log which was a text file. The first step of data preprocessing was to filter and clean up the data. The next step of data preprocessing was to define process instances (cases). Our approached used a document as a notion of process instance. We utilized the concept of perspective, proposed by Song et al. [13] to partition event sequences. Our perspective of the event data log was based on documents. Particularly, we wanted to find out how users interact and coordinate for writing and reviewing documents. The final step in data preparation was to transform the log file to a standard format for process mining. Process mining tools such as ProM use MXML (as in *Mining XML*) files as sources [15]. The transformed MXML file was then used as a source for a process mining tool like ProM.

3.2 Log inspection

After preprocessing, the resulting event log consisted of 29 documents with a total of 4,677 events. Each process case represented one document. There were 8 different types of events (Section 3.4 described the process model and event types). The bar chart of Figure 3 shows the number of events for each of the 29 documents, represented by the length of the bar (*DocXX* denotes the document of *Group XX*). The documents are ranked based on their final mark ranging from 4/10 to 10/10. For example, *Doc07*, *Doc08*, *Doc09*, *Doc10*, *Doc14*, *Doc17*, *Doc21*, *Doc27* and *Doc29* all obtained the highest mark, i.e. 10/10, while *Doc11* obtained the lowest mark of 4/10. On average there are 161 events per document. The maximum number of events is 369, with *Doc12*. *Doc10* has the smallest number of 45 events associated with it.

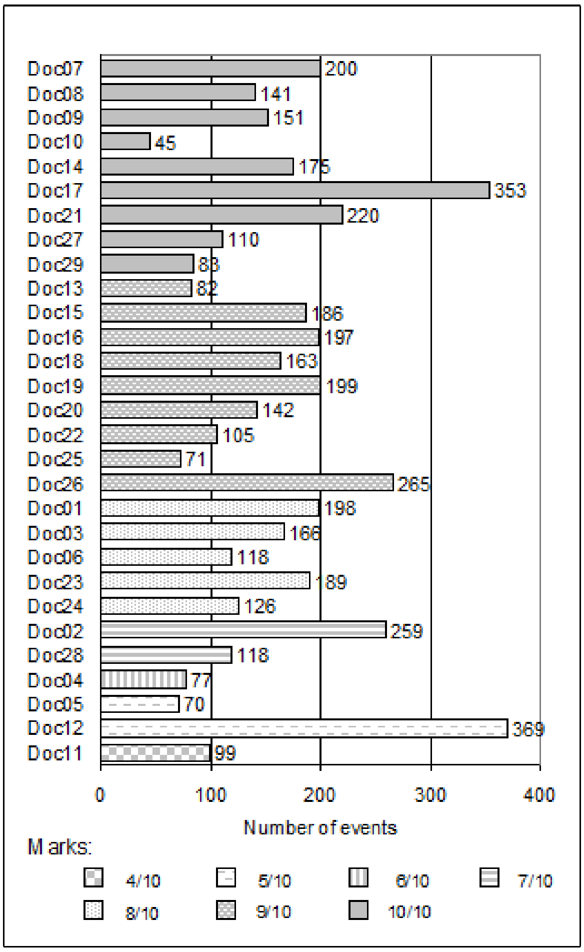

Figure 3: Comparing number of events of 29 documents ranked by their final marks.

Based on the number of events presented in Figure 3, we could not distinguish the better from the weaker groups. Although Group 12 has the maximum number of interaction events, it was ranked in the 6th place. In contrast, the document of Group 10 with the least number of interactions was given the highest mark. In addition, simple statistics drawn from the figure could not clearly provide understanding of students' interaction. Therefore, further analysis was made in order to distinguish group performance and cooperation. We will describe it next.

3.3 Historical snapshot of reviewing activities

The Dotted Chart Analysis utility of ProM [10] was used to analyze students' reviewing activities. The dot-

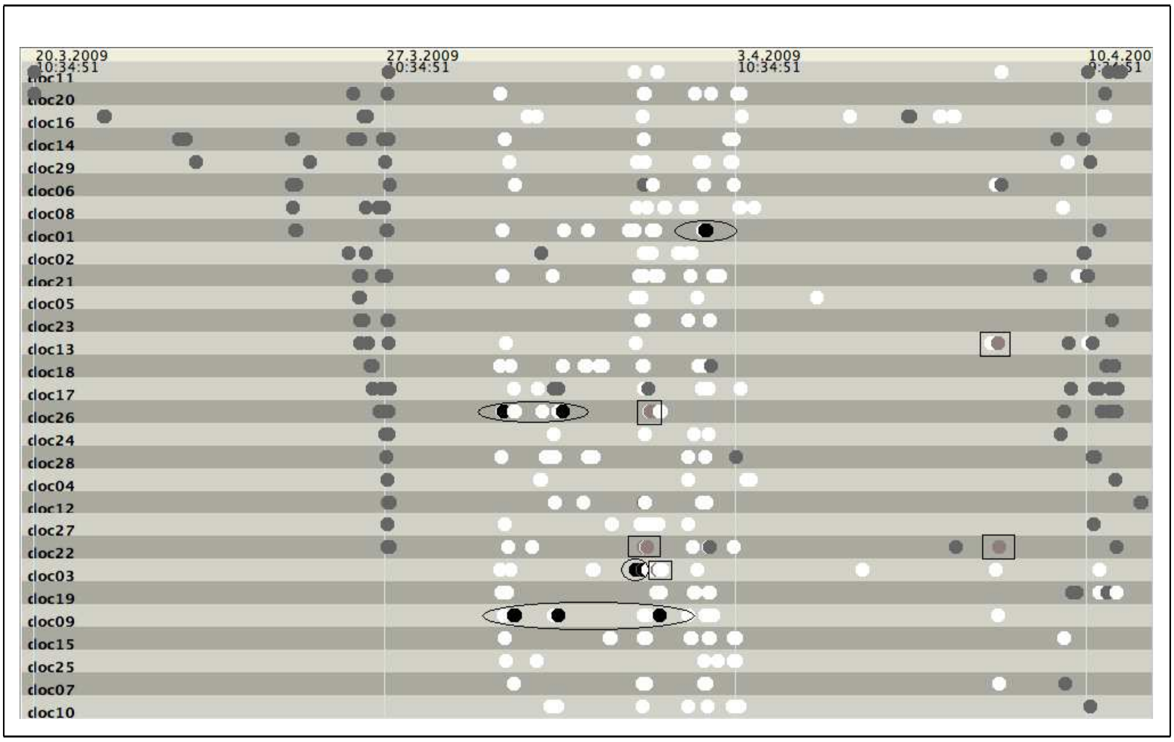

Figure 4: Dotted chart of 29 reviewed documents ordered by their first events' timestamps (from ProM tool [10]). Grey denoted events generated by by authors; white by reviewers, black by reviewers' group member (indicated by ovals) and brown by others (indicated by rectangles).

ted chart is similar to a Gantt chart [14], showing the spread of events over time by plotting a dot for each event in the log. Figure 4 illustrates the output of the dotted chart analysis of students' interaction for reviewing their PSD documents. All instances (one per document) are sorted by start time (the first event ever happening for a particular document during the three-week usage of the system). As shown in the figure, there are three important dates due to the three compulsory submissions: PSD for peer review on 27th March 2009; feedback of peer review on 3rd April 2009, and final PSD on 10th April 2009. In the figure, points represent events occurring at certain time. For particular documents, different color denotes events generated by different roles of users: grey events generated by authors, white events by reviewers assigned for peer review, black events (circled in the figure) by team members of assigned reviewers, and brown events (shown in rectangles) by *non-author* users who were neither assigned reviewers nor assigned reviewers' team members.

We can clearly see from the figure that 22 documents have been revised using Glosser in the first week before the submission for peer review. Obviously, those documents were only used in the system by the authors as indicated by grey events. There were 7 documents starting in the second week. They belonged to groups: 7, 9, 10 (received the same marks of 10/10); 15, 19, 25 (9/10); and 3 (8/10). This means that these documents have never been revised by their authors using Glosser before submitting for peer review. Nevertheless, these seven documents received high marks in the final assessment.

In addition, we observed that all activities of peer review happened in the second week before the submission of feedback. Most of the reviewing activities were performed by the assigned reviewers as indicated by white dotted events. This met the intention of the course of using Glosser for peer review. There are two interesting types of events in Figure 4. Firstly, 4 documents have events originated by students who were not the authors nor the assigned reviewers, as can be seen by black dots of documents of groups: 9 (received a mark of 10/10); 26 (9/10); and 1, 3 (8/10). Those events suggest that students either assisted their team members to review their assigned documents or performed the peer review task together with their group members sitting side-by-side using only one account. We discussed this matter with the course instructor who was also aware of this problem and will try to find a solution to prevent this problem happening in the next semester. Secondly, there are a small number of events where students reviewed others' documents which were not assigned to them nor to their team members for peer review, as indicated by brown dotted events for documents of groups: 3, 13, 22, and 26. These documents received good marks ranging from 8/10 to 9/10. We believe this happened when students shared their own PSD to their friends to assist them using Glosser. We will perform further investigation to prevent this case from happening next year.

In addition, from Figure 4 we can notice that eight different documents were not revised by their authors using Glosser before the final submission. These documents were 8, 9 (10/10); 15, 16, 19, 25 (9/10); 3 (8/10); and 5 (5/10). Except document of group 5, all documents received the top three highest marks. In fact, three of them (9, 15 and 25) have never been revised by their authors using Glosser at all. This implies that the better groups used feedback received from peer review and the instructor as main source for revising their PSD. They did not spent much time using Glosser for revising their own documents. It is also interesting to note that reviewing activities did not evenly spread out for the three-week period of running the system. In fact, the system has only been used extensively for peer review in the second week as we can see in the figure. There were not many interactions in the third week. However, a number of activities happened a few days before the final submission.

To sum up, the dotted chart tool in ProM allows us to analyze reviewing activities in order to seek information on how each of 29 documents was reviewed by groups of students with different roles. We further investigated patterns of students' interaction for reviewing those documents, as described in the next subsection.

3.4 Process discovery and sequence analysis

From the event log of our case study data, we extracted the process model shown in Figure 5, which represents the process common to all the groups. Groups began with events of opening a particular document (ROD). Then, the reviewing tool was requested (TOR). After that, different reviewing activities were performed and the resulting feedbacks were displayed. The process reiterated until users logged off or closed their browsers. As discussed in Section 2.2.2, the reviewing activities involve these tools: HOT, FLT, KTG, GRT, TOT, and KTH.

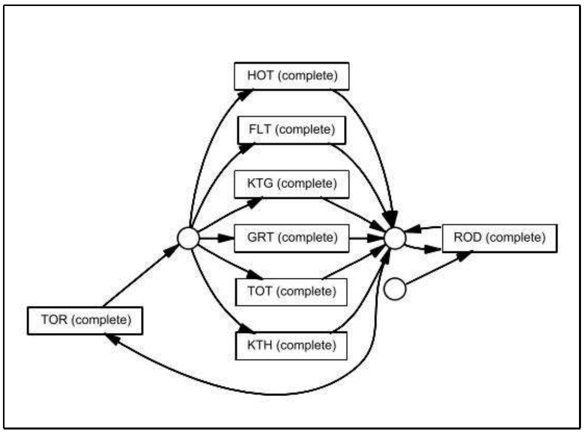

Figure 5: Process model of usage of the reviewing tool, Glosser.

We were naturally interested in finding out more about individual group activity and the path each group was following in this process. ProM provides a Performance Sequence Analysis plug-in to find the most frequent paths in the event log [2]. Figure 6 illustrates the interaction for documents of two groups in the course, with Group 1 (received a mark of 8/10) at the top and Group 29 (10/10) at the bottom. All eight events represented on horizontal axis are according to events discovered by the process model mentioned above. We examined sequence patterns for all documents of 29 groups. We discovered that only one document (*doc10*) was not used with all reviewing tools. In fact, the authors and reviewers of the document only utilized the HOT tool.

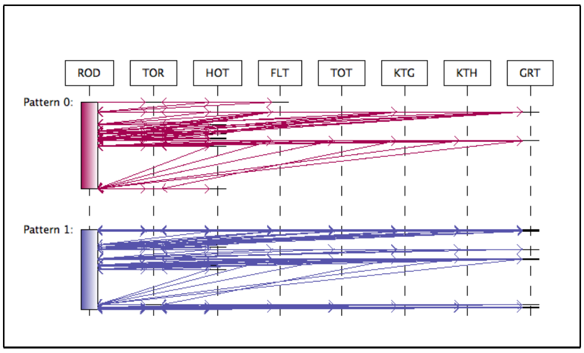

Figure 6: Sequence analysis of documents of Group 1 (above) and Group 29.

We have also used the same plug-in to extract all reviewing interactions for each document. This further investigation gives an insight on how different users reviewed documents using Glosser. For instance, Figure 7 depicts the users' interactions of two documents (*doc01* on the right and *doc29* on the left). Each column represents a user, where G29-1 is user 1 of Group 29 and so on. We analyzed all documents and found that more than half of them were revised by only one author using Glosser. In other words, although students worked in groups, only one member actually performed the reviewing task using the system.

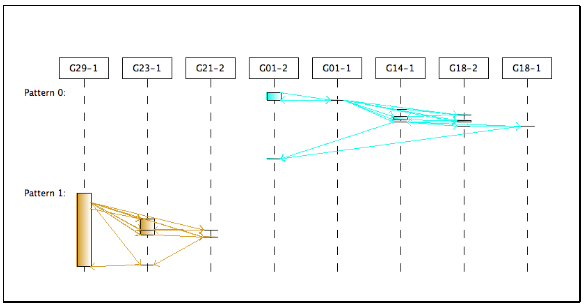

Figure 7: Users' interaction of Group 1 (right) and Group29.

In this section, we illustrated the potential of process mining techniques in understanding how writers react to peer review feedback and to the use of an automatic feedback tool like Glosser. In the log preparation, our notion of process instance is based on documents because we would like to analyze user interactions on a same document. Dotted chart and sequence analyses were used to gain insights on how students reviewed their documents.

4 Discussion and conclusion

The work described here is a work in progress. While the case study presented here illustrates how our framework, WriteProc can be used, the data we used did not allow us to discover sequence patterns correlated to better outcomes. Although a pattern of users' interaction can be extracted for a particular group, there are different patterns for different groups. Indeed, we could not draw a significant pattern among groups in order to distinguish the better from the weaker groups. However, this gave us direction for the next step of our work. One way to improve our understanding of what writing processes lead to better outcomes so software tools can be used to provide advice during the writing process, is to use text mining techniques. For collaborative writing, we would like to have insights on how each version of documents changes in order to understand the writing process of each document. Although we are able to track all versions of documents that were reviewed in the system, this tracking analysis does not yet give us meaningful insights about the purpose of the text changes between each version. One possibility to systematically capture and interpret writing activities in collaborative writing is to understand changes in text written in each version of the document. Currently, we are working on extracting changes in concepts and ideas during the writing of documents. The text mining algorithms use vector representations of the documents accounting for the temporal nature of the data and the character of writing interaction. The result of the text mining tool will be analyzed and combined with the outcome of process mining (like the one described in the current case study).

Based on the process mining tool illustrated in the case study and text mining techniques as described above, we are developing WriteProc to provide visualization depicting users' interaction and collaboration in order to support writing activities. For example, a user interface can be built to assist a group of writers in identifying a plan for their writing process. This plan is created at the beginning of writing process representing a master plan of all writing activities and tasks. At particular point in time, writers can specify which stage they are on their writing process. During a time of writing, the system monitors if current writing activities are according to the writer's specification. For instance, a leader of a group of writers assigns all writing tasks to his or her members. The group leader specifies that the group is currently drafting its documents. WriteProc will track the group's writing activities and perform semantic analysis of the written texts. If it finds out, for example, that the members are actually outlining the documents (instead of drafting), it will provide information about their writing activities as feedback to the group. In this case, the writers can either adjust and modify their writing process specification, or investigate and change their written content according to the feedback given by the system.

In conclusion, we contribute here the description of WriteProc a framework that combines process and text mining techniques. The architecture of the system is described together with its integration to Google Docs as an environment for users to do the actual writing, and to the Google API that allows the tool to collect the revision information. A case study with a real teaching scenario is described and used to show how the tool can be used to analyze the process component of a collaborative writing task.

Acknowledgements This project has been funded by Australian Research Council DP0986873. The authors would like to thank Jorge Villalon and Stephen O'Rourke for their help with Glosser and TML.

References

[1] N. O. Andrews and E. A. Fox. *Recent Developments in Document Clustering*. Technical Report TR-07-35, Computer Science, Virginia Tech, 2007.

[2] M. Bozkaya, J. Gabriel and J. M. van der Werf. Process diagnostics: A mothod based on process mining. In *International Conference on Information, Process, and Knowledge Management*, February 2009.

[3] P. A. Carlson and F. C. Berry. Using computer-mediated peer review in an engineering design course. *IEEE Transactions on Professional Communication*, Volume 51, Number 3, pages 264–279, Sept. 2008.

[4] K. Cho and C.D. Schunn. Scaffolded writing and rewriting in the discipline: A web-based reciprocal peer review system. *Computers & Education*, Volume 48, Number 3, pages 409–426, 2007.

[5] C. A. Ellis, K. Kim and A. J. Rembert. Workflow mining: Definition, techniques, and future directions. *Workflow Handbook*, pages 213–226, 2006.

[6] A. Goldberg, M. Russell and A. Cook. The effect of computers on student writing: A meta-analysis of studies from 1992 to 2002. *Journal of Technology, Learning, and Assessment*, Volume 2, 2003.

[7] E. Hatcher and O. Gospodnetic. *Lucene in Action*. Manning Publications Co., 2004.

[8] P. B. Lowry, A. Curtis and M. R. Lowry. Building a taxonomy and nomenclature of collaborative writing to improve interdisciplinary research and practice. *Journal of Business Communication*, Volume 41, pages 66–99, 2003.

[9] S. O'Rourke and R. A. Calvo. Semantic visualisations for academic writing support. In Vania Dimitrova, Riichiro Mizoguchi, Benedict du Boulay and Art Graesser (editors), *14th Conference on Artificial Intelligence in Education*, pages 173–180. IOS Press, July 2009.

[10] ProM. Version 5.2. `http://prom.win.tue.nl/tools/prom/`, 2009.

[11] M.D. Shermis and J. Burstein. *Automated essay scoring: A cross-disciplinary perspective*, Volume 16. MIT Press, 2003.

[12] J. Song, T. Luo and S. Chen. Behavior pattern mining: Apply process mining technology to common event logs of information systems. In *IEEE International Conference on Networking, Sensing and Control*, Sanya, April 2008.

[13] J. Song, T. Luo, S. Chen and Feng Gao. The data preprocessing of behavior pattern discovering in collaboration environment. In *IEEE/WIC/ACM International Conferrence on Web Intellengence and Intenllent Agent Technology*, pages 521–525, Silicon Valley, USA, November 2007.

[14] M. Song and W. M. P. van der Aalst. Supporting process mining by showing events at a glance. In *7th Annual Workshop on Information Technologies and Systems*, pages 139–145, 2007.

[15] B. F. van Dongen, H.M.W. Verbeek A. K. A. de Medeiros, A. J. M. M. Weijsters and W. M. P. van der Aalst. The ProM framework: A new era in process mining tool support. *Lecture Notes in Computer Science: Application and Theory of Petri Nets*, pages 444–454, 2005.

[16] J. Villalon, P. Kearney, R.A. Calvo and P. Reimann. Glosser: Enhanced feedback for student writing tasks. In *The 8th IEEE International Conference on Advanced Learning Technologies*, Santander, Spain, July 2008.