Assisting Instructional Assessment of Undergraduate Collaborative Wiki and SVN Activities

Jihie Kim, Erin Shaw, Hao Xu, Adarsh G V
Information Sciences Institute / University of Southern California
4676 Admiralty Way, Marina del Rey CA 90292

{jihie, shaw, hxu, agvenkat}@isi.edu

ABSTRACT

In this paper we examine the collaborative performance of undergraduate engineering students who used shared project documents (Wikis, Google documents) and a software version control system (SVN) to support project collaboration. We present an initial implementation of TeamAnalytics, an instructional tool that facilitates the analyses of the student collaboration process by creating dynamic summaries of team member contributions over time in. Document content is processed using machine learning techniques. We validated the summary's effectiveness using a questionnaire given to instructors and team managers. Team managers indicated that summaries of student contribution to coding activities influenced their evaluation and coordination of team projects.

Keywords

Collaborative teamwork, instructor tool, wiki, team projects.

1. INTRODUCTION

Engineering students participating in collaborative activities communicate electronically through a variety of applications, most of which are inaccessible to an instructor and thus offer little insight into the process of collaboration. The goal of the Pedagogical Wiki project is to assist instructors and educational researchers in evaluating team and individual student performance in the context of computer-supported collaborative learning environments.

In this paper we examine the collaborative performance of undergraduate engineering students who used a shared project documents, including Wikis and Google documents, and a software version control system to support project collaboration. Wikis are editable Web sites that support the creation of linked pages, archiving of media, revision control, access control, searching, and a consistent look and feel. Wikis facilitate collaborative learning by allowing groups of laypersons to collaboratively create web content [13,1,4]. However, the research on the effectiveness of using Wikis for student collaboration has been mixed [14,17], and patterns of student collaborative documenting and their effect on learning have not been fully assessed.

In addition to Wikis, students used Google documents, a popular team document generation and sharing environment that allows synchronous document editing, and Subversion (SVN), a version control system that is commonly used for software management. Version control systems track revisions that are made to files over time, usually by a group of authors. Wikis, Google documents and SVN all provide revision "histories", which can, in theory, be used to analyze student performance. For example, Ben-Zvi [1] notes that while logs can be used to evaluate each student's Wiki contribution, the number of contributions is enormous and new techniques and tools are needed to track them efficiently. Without proper tools, the analysis of document histories would place a considerable burden on instructors, who rarely have the skills or time to analyze the data for assessment purposes.

This paper presents a new instructional tool called *TeamAnalytics* that summarizes collaboration via online team activities. It dynamically processes student shared document edits and code management actions, summarizes both the overall team and individual contributions in each week, and presents the summary to the team managers and the instructor. For processing Wiki content data, we use natural language processing (NLP) techniques and machine learning approaches to generate topic-based summary of the documents. We report a study of PedSummary based on team manager ratings and a small survey. The initial results with two undergraduate courses with large team projects indicate that individual code contribution summary is useful for team managers and such summary can influence how the managers coordinate the team project.

1.1 Teamwork summary categories

Table 1. Current categories of team work summaries.

	Category	Summary	Description
1	View Docs	A tree view of Wiki docs	Hierarchical view of the docs, organized based on links and topics
2	Wiki-Group	Total Wiki contributions by members	Number of docs (and number of words) created, frequencies of accesses/views, and topic-based document distribution
3	Wiki-Individual	Individual Wiki contributions	Number of docs, words, access/view of others' docs and topic-based distribution of the docs
4	SVN-group	Total SVN contributions by members	Number of files added/modified/deleted by the team, and weekly totals
5	SVN-Individual	Individual SVN contribution	Number of files added/modified/deleted by the student, and weekly totals

As engineering researchers, we (the authors) use Wikis extensively, primarily as a knowledge repository for project documentation and media. It is clear that, for Wikis, the benefit of democratic use is also its downfall, with its lack of structure and

oversight. Student Wiki sites often do not scale well, and tracking text and asset contributions becomes frustrating. Our goal was to provide finer-grained measurements and user-friendly interfaces for understanding instructional shared Wiki use.

In order to alleviate the problem of viewing student documents with existing Wiki systems, TeamAnalytics clusters documents according to the link hierarchy within the Wiki system (Category 1 in Table 1). They are also organized based on document topics that are automatically identified from topic models.

The primary instructor for the undergraduate course we studied also teaches an upper level course in which the students take on the role of team managers for the undergraduate project teams, and the instructor delegates most of the assessment tasks for the project teams to the team managers. A needs assessment was performed for both the course instructor and the team managers. Although the team managers participate in student group meetings and help the students as needed, often times they had difficulty in documenting who is doing what and how much. Such manager documentation is used in reporting teamwork to the instructor and tracking the teamwork throughout the project. In some cases, the members in the same team receive similar grades depending on the team performance. The instructor and the managers wanted to see individual contributions as well as the total contributions by all the team members (Categories 3 and 5 vs. 2 and 4 in Table 1). Identifying patterns of student activity relative to student performance was also discussed. In order to support an analysis of activity patterns, we broke up the contributions into weekly activities so that the managers can see how students work towards the deadlines over time.

2. ANALYSIS REPORT GENERATION

This section describes how collaborative Wiki and Google document activities are captured into a summary that is viewable by team managers and instructors. Although we show results for Moodle's Wiki, Google documents and SVN, most of the data processing steps do not depend on the course management system or particular document tools. For example, our topic classification functions are being used for other Wiki (e.g. Brainkeeper) content.

2.1 Participating Courses

The TeamAnalytics system was integrated into Moodle's [10] virtual learning environment during the Spring 2011 and Fall 2011 semesters. During each of these semesters, two undergraduate software engineering courses were combined for a large team-based project assignment. The study took place at the University of Southern California.

Students in a freshman level software development course (CSCI200) teamed up with students in a sophomore level course (CSCI201) for a large-scale programming project. Students in both courses learned team management, software engineering principles, and operating system principles and used the concepts to build "authentic" applications that solved new problems. Because second year students had already completed the first year course, they were able to mentor the first year students. The project team had students from both classes. Each team had about four freshmen and four sophomore students. The first year course (CSCI200) emphasized user-interfaces and the second year course focused on architecture (CSCI201). Additionally, a team manager was assigned to each team to assesses team co-ordination and

leadership skills, and provide help throughout the project. Our work focused on assisting the team managers and the instructors.

There were ten teams of between ten and fifteen students each semester. The teams used their collaborative workspaces (Moodle) in myriad ways. Some teams used the Moodle Wiki and some used Google documents that they then linked to the Moodle courses. Some used a combination of both, e.g. Wikis for meeting notes and Google for documents. The choice was theirs. The workspace for team M2, is shown in Figure 1.
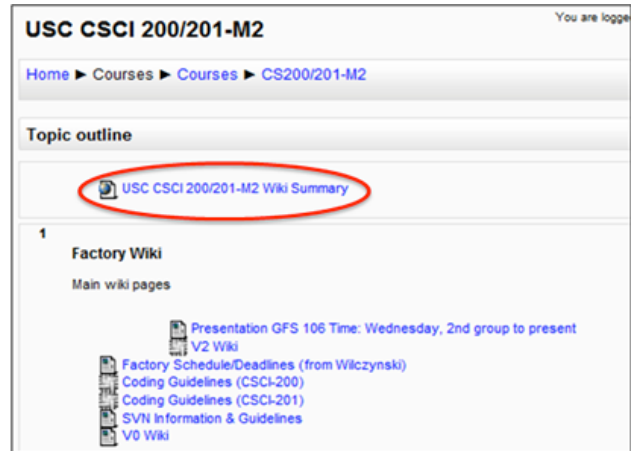

Figure 1. The collaborative workspace for a combined USC freshman/sophomore engineering team M2

2.2 Data processing

The TeamAnalytics architecture is shown in Figure 2. All team activity data is stored in the Student Group Activity database. The system fetches SVN activity data from the SVN server used by the courses. Students' actions including addition, modification and deletion of files are retrieved every 24 hours.

The system also dynamically accesses the student Wiki history including addition, deletion and modification information from the course management system. Each team provided edit permissions to allow us to access to the content and edit history of the shared Google documents through a Google API. After re-formatting the data, the Wiki data processing functions were used.
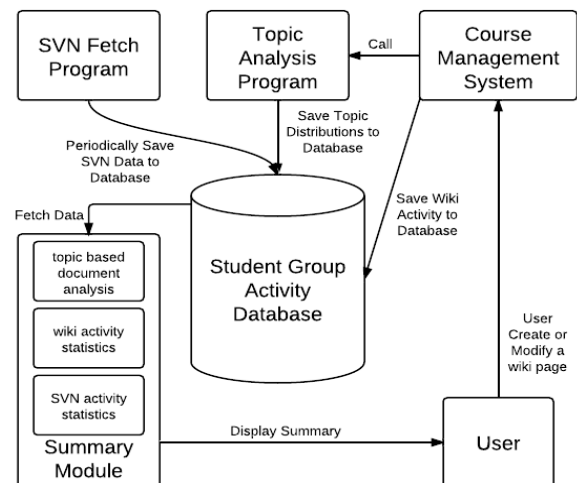

Figure 2. Generating teamwork summary using data from SVN and Wikis and Google documents.

For topic modeling, whenever a new page or a revised page is saved, a backend program is invoked to parse the content of the page and generate topic distributions using the automatic topic classifier, which is described in detail in subsequent sections.

The dynamically generated summaries were sent to the team managers and instructors by weekly email. The summaries were also viewable from within the team's Moodle course environment. The team manager of M2 could access the summary by clicking the 'USC CSCI200/201-M2 Wiki Summary' link (Figure 1). The instructor and the team managers could view all the teams' activities as shown in Figure 3. The content of the summaries is described in Section 3.

Figure 3 Summaries of all the participating teams available for the instructors and team managers.

2.3 Automatic topic classification

The Wiki pages and Google docs are classified based on the page title and the content using Labeled LDA.

2.3.1 Background on Labeled LDA

Because we wanted to develop a topic modeling approach that could be easily applied to different courses, supervised approaches requiring a large amount of labeled data were not appropriate. And because discussion datasets are noisy, we needed a model that could capture semantic meanings behind the words rather than words themselves. LDA (Latent Dirichlet allocation) 32] is very powerful in analyzing latent topics of documents, but it has all the disadvantages inherent to any unsupervised model. The topic distribution of LDA depends on the word distribution in the documents and cannot be controlled even if we have a prior knowledge to guide topic generation. Thus many topics are just a cluster of words that co-occur in many documents and do not have a semantic meaning in real data. Ramage et al [13] introduced a semi-supervised algorithm, called Labeled LDA, a novel model that uses multi-labeled corpora to address the credit assignment problem. Unlike traditional LDA, Labeled LDA constrains topics of documents to a given label set. We have V number of unique vocabularies and D number of documents, and K number of topics. For each document d, which consists of a list of word $(w_1(d),…,w_N(d))$, we have k dimensional binary topic indicators. Unlike using symmetric Dirichlet distribution with a single hyper parameter α as a Dirichlet prior on the topic distribution $\theta(d)$, Labeled LDA restricts $\theta(d)$ to only over the topics that correspond to observed labels. The key task was to select a label set that could generate meaningful topic results.

2.3.2 Wiki Topic Modeling with Labeled LDA

The topic categories for the software engineering team wiki documents are shown in Table 2. This was generated after manual analyses of the course curriculum and the content of the wiki documents across all the project groups in the class. The topic categories represent the major types of the documents generated by the students over the course. The two main topic classes are team management categories (Team Organization and Progress Summary). The rest of them represent software engineering principles documents that show Initial Planning, Design, Coding, Testing and System Analysis.

Table 2. Topic categories for team work document.

Topics	Description	Kappa
Team Organization	Team contact information, availability for meetings, allocation of project modules to members, milestones and their target dates.	0.83
Initial **Planning**	Initial research and Q&A discussions on the project. Some use external links and reference materials.	0.82
Design	Design documents describing frontend, backend, Interfaces. Also includes interactive diagrams.	0.73
Implementation and **Coding**	Describes implementation method, using pseudo code or java code.	0.91
Testing	Describe code testing including unit test cases, bugs or Junit.	0.73
System **Analysis**	Analysis of implemented system; system issues, setbacks and fixes in system design.	0.74
Progress Summary	Document discussions, meetings, email and phone conversations on the status of project modules.	0.81

A Kappa measure [5] was used to verify agreement. Table 2 shows the Kappa values between two annotators for 263 documents sampled. Kappa values take into account agreement that can occur by chance.

$$\text{Kappa} = \frac{\text{Observed agreement - Chance agreement}}{\text{Total observed - Chance agreement}}$$

Table 3. Sample label set and LLDA classification results.

Topics	Sample Labels	%	Prec/re call
Team Org	rxgui,rxnub,part,kit,on,panel,gantri,rxnub_rxnub,rxfunction,object,return,user,factori,move,can	14.8	0.4 /0.29
Planning	rxquestionmark,custom,=,{,},messag,if,rxfunction,order,state,list,check,action,customer,timer	3.5	0.0 /0.0
Design	kit,part,public,rxnub,rxfunction,int,void,on,stand,rxquestionmark,type,public_void,call,=,at	28.3	0.43 /0.35
Coding	rxfunction,=,part,{,rxnub,},if,void,int,public,kit,set,bin,rxgui,feeder	19.0	0.69 /0.82
Testing	},{,part,=,kit,rxfunction,if,nest,int,return,rxquestionmark,lane,void,true,list	9.4	0.0 /0.0
Analysis	rxgui,you,panel,rxnub,food,cook,on,ar,when,anim,should,do,if,all,your	6.3	0.54 /1.0
Progress	rxnub,rxnub_rxnub,i,test,meet,task,rxday,my,integr,code,on,rxgui,done,design,panel	16.7	0.65 /1.0

Sample label sets used for LLDA are show in Table 3. We evaluated the model distributions using the manual annotations as the gold standard. Since documents can contain multiple topics, we evaluated them by selecting and comparing the top 2 topics

from the manual annotations and model results. Precision is defined as the ratio of the number of correct topic annotations generated by the model to the total number of topic annotations generated by the model. Recall is defined as the ratio of the number of correct topic annotations generated by the model to the total number of correct annotations specified by the gold standard. The table also shows the % of the topics within 314 annotated documents. The current model provides limited accuracies for some topic categories due to limited examples. We are currently improving the LLDA model by adding more dataset.

3. ANALYSIS REPORT PRESENTATION

This section describes how document-based and code-based activity summaries were presented to team managers and instructors. As described above, the dynamically updated summaries and statistics were viewable from within Moodle. We also generated and sent team summary reports to each team manager by email.

3.1 Document Summary

This section describes the content of the document summaries.

3.1.1 Tree view of document with topic labels

A tree view of the documents created or modified by students on team W3 is shown in Figure 4. Each team generated more than hundred documents and uploaded many additional files such as design diagrams. Wiki pages, plan text pages, and upload documents of any type were stored within the virtual learning environment. Wiki pages were related using hyperlinks. Google documents were also used and linked within Moodle. In order to help students and team managers navigate through various documents, TeamAnalytics compiled document links and generated a hierarchical view of the team documents. A general API (application programming interface) was developed so that other types of links could be captured within the structure.

The tree view also shows who created the document, how many students edited the document, how many edits were made, how long the document was edited, how many words were included, and how many links were present in the document. We also organized the documents based on the content topics, using the above-mentioned LLDA models. Without reading the individual document details, team managers could evaluate who was contributing on what topic and how often.
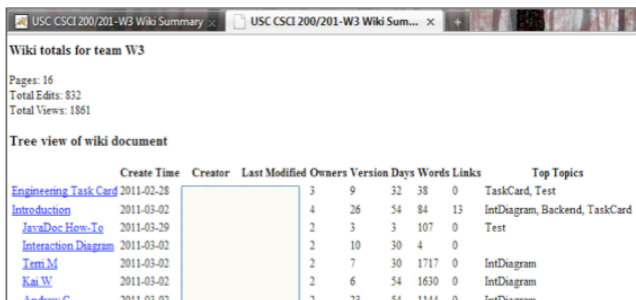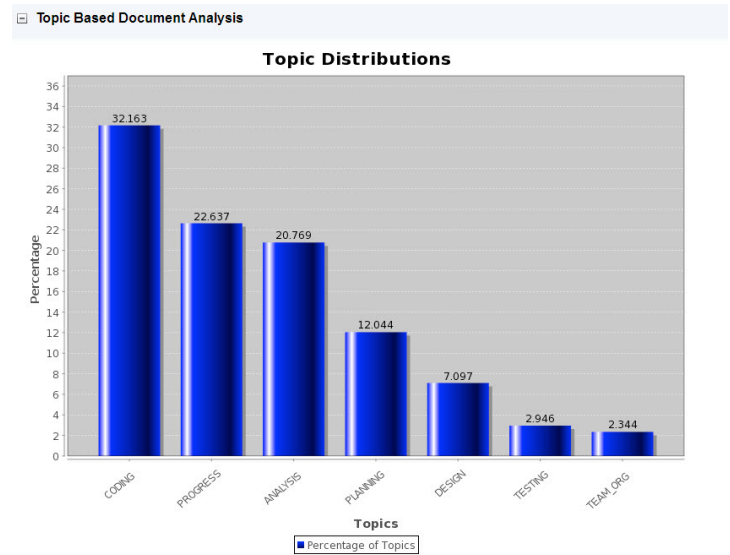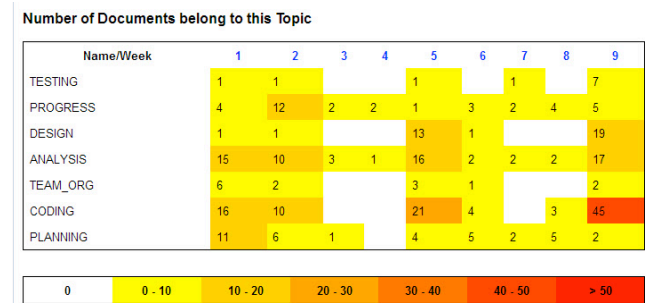

Figure 4. Tree view of documents based on document links

3.1.2 Topic based document distribution

(a) Topic distribution of all the team documents.

(b) Weekly document topic distribution of documents.

Figure 5. Topic distribution of team documents.

Document topics were summarized into a bar graph like the one shown in Figure 5-(a). The accumulated number of documents per each topic, based on the LLDA topic distribution, is shown. Using this view, the team managers could estimate the distribution of topics in the team documents. We also highlighted increments within a given week so the viewer could evaluate the topics of focus during that week. A weekly distribution of the document topics is shown in the heat map in Figure 5-(b). The headings 1-9 depict the nine weeks that the project runs. The cells with high frequency values are highlighted with darker colors.

3.1.3 Participation frequency per student

Wiki contributions by individual students are shown in Figure 6. For each student the left (blue) bars show number of documents viewed and the right (green) bars show the number of documents edited by the student. The portions contributed during the current week are highlighted with lighter colors, and the counts at the tops of the bars show the current week's numbers of edits and views.
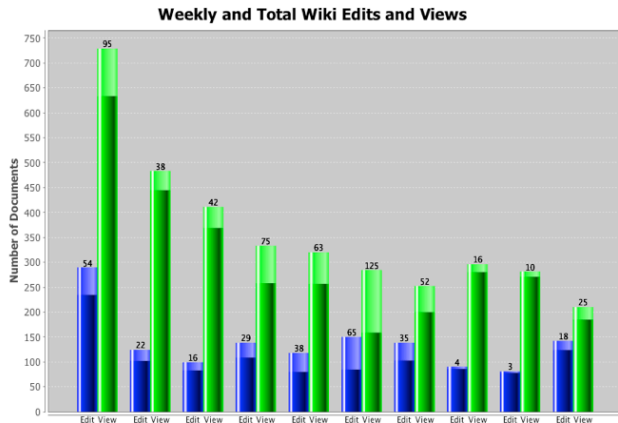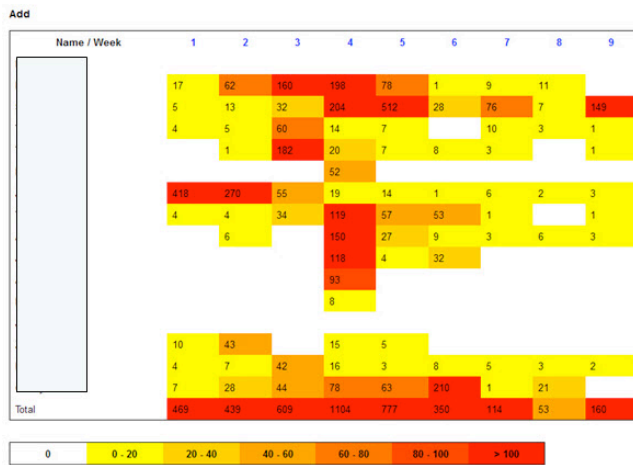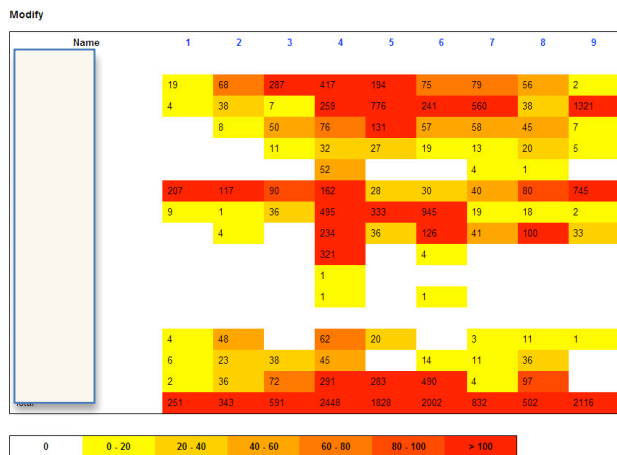
Figure 6. Individual student contributions to Wiki

3.2 SVN Summary

(a) Number of files added by each student weekly.

(b) Number of files modified by each student weekly.

Figure 7. Weekly student contributions to SVN

Students used the Subversion (SVN) version control system to manage changes to their team's programming files. SVN allowed team members to add new program files, or modify or delete existing ones. Figure 7 shows individual student contributions to SVN for adding and modifying files. The weekly total numbers of file additions and modifications by all the team members are shown at the bottom of the table. The team managers were able to track the degree of SVN activity using this summary.

4. USER STUDY

TeamAnalytics was integrated into Moodle's virtual learning environment during the 2011 Spring and Fall semesters. A total of 278 students participated in the projects (42 freshmen and 67 sophomores in the Spring implementation, and 90 freshmen and 57 sophomores in the Fall implementation). There were ten teams each semester, and a manager was assigned for each team. The system was introduced to the classes and team managers before the project started. The dynamic summary was available to team managers on Moodle, and also sent weekly by email.

Table 3. Team manager ratings of the TeamAnalytics components.

Team Manager Rating (average)	Spr 2011 (N=7)	Fall 2011 (N=7)
Viewing: 1=never, 2=occasionally, 3=often		
Document activity viewed	2.0	1.71
Topic activity viewed	--	1.57
SVN activity viewed	2.7	2.29
Helpfulness: 1=minimal, 2=moderate, 3=very		
Document activity helpful/ influence manager	1.9	2
Topic activity helpful/ influence manager	--	1.6
SVN activity helpful/ influence manager	2.4	2.8

Survey responses for the team managers are shown in Table 4. Survey participation was voluntary and the response rate for both semesters was seven out of ten. Team managers were asked to rate the document (Wiki and Google Docs) activity views, topic-based document summaries and SVN activity summaries separately. The topic-based document summary was developed later and introduced to the Fall 2011 classes only.

The team managers viewed the SVN activity summaries more often than the document and topic summaries, and found the SVN summaries between moderately and very helpful. The documents summaries were rated moderately helpful and the topic summary was rated between minimal and moderate. It is evident that the team managers were most interested in student coding activities.

Team manager responses to other survey questions are shown in Table 5. The managers liked how they could keep track of coding work progress using TeamAnalytics. Several managers raised issues about the user interface (UI) especially comparing the old and new Moodle UIs. Recent upgrades and our own improvement of the interface design reduced some of the concerns. Individual managers show different preferences for how the information should be presented. We are investigating alternative approaches for showing the results. The managers also wanted to see more

details on student coding activities such as the numbers of lines added or deleted by individual students. We plan include such coding activity information and provide a drill-down view where end users can choose to see such details

Table 5. Team manager answers for survey questions.

Question		Responses
How did the summary influence your work?	s p	showed me the progress of the team
		It did influence the way I saw the member and the team as I knew who was putting in more effort. This is definitely better than the views and edits
		I will review and evaluate team members' performances for their projects.
		Frequency of SVN commits does not map to work done on the project, but low commits CAN be a reason for concern
		Allowed for me to monitor which members were contributing to code.
Can you think of additional activity information that you want to see?	s p	Perhaps the time of the last edit or view
		Member wiki contribution statistics
		I like it how it is
	f a	If it's possible, analysis showing number of lines of code affected, number of files touched, whether the commit is a merge. Any of those would be helpful for analysis
		I collected activity data by being at the meetings, so no.
		Meeting set up and attendance checking.
		Individual doc revision statistics for each document
		The question of are we behind, maybe by comparing with SVN commits of previous semesters?
		Whether someones work has actually been productive
Do you have any suggestions for improving the summary, including how the information is displayed?	s p	The current way is good enough as it serves the purpose, The UI should be improved
		In graph instead of tables. Better interface
		moodle's UI sucks
		you guys were great with feedback this semester and the moodle was FAR more useful than it was in the fall of 2010. I like it how it is.
		In a dashboard format where priority is given to certain issues. Bug tracking.
		Information would be more easily shared if it was in more of a forum type setup.
	f a	Allow more detail in the SVN log, like separate it by day or hour (on a graph) instead of week. Also stronger analysis like graph of SVN activities for each person.
		Less colored charts, more bar graphs

5. RELATED WORK

Our work is situated in the research domain of context modeling and activity awareness to support group performance on complex tasks (e.g., [3,18]). Of particular relevance is Upton and Kays' Narcissus system [16], which graphically models user and group behavior to support team collaboration. Also related is Suthers, Dwyer and Medina's [15] Uptake Analysis Framework for conceptualizing and representing distributed interaction, in which contingency graphs are used to transcribe activity in the temporal space, distributed across multiple documents, to enable researchers to (possibly) identify the influence of prior activity on ongoing activity. Our work extends existing research by automatically generating summaries of group work in collaborative knowledge building and team programming environments, and by combining NLP techniques to support topic-based analysis of contribution content.

Our work builds on Activity Theory [6,9], which we used previously as a framework for analyzing wiki activity [7]. The presented work significantly extends the scope of activity analyses and presents an evaluation with team managers. Glassman and Kang [8] propose that learning via Wikis and Web browsing is best explained as an abductive logic process, consisting of discovery and hypothesis generation, which would call for a model that reasons about prior activity to explain ongoing activity. The TeamAnalytics effort facilitates analysis of student online work contributions and how they progress over time by instructors and team managers.

TeamAnalytics also extends our prior work on workflow-based analysis of student online discussions [11,12]. We plan to make use of the computational workflow framework to support more efficient and robust approaches for assessing student online activities.

6. SUMMARY AND FUTURE WORK

This paper presents our initial implementation of TeamAnalytics that provides a summary of member contributions over time in Wiki space and SVN. Our initial study with team managers indicates that a summary of how individual students contribute to coding can influence how the managers evaluate and coordinate the team project. We plan to trace how the managers use the information in team coordination and assisting students. We will also explore opportunities to assist grading student teamwork with the TeamAnalytics report.

Based on the team manager comments collected so far, we plan to add more details on student coding activities including whose files were modified by whom. We are also investigating additional topic categories that can help instructors and managers track student activities. In order to receive more feedback while the team managers view the summaries, we plan to add feedback fields in the summary page so that we can capture team manager input regularly. Although the instructors do not directly manage teamwork, they can also provide input on how to make the summary more useful using such function.

Regarding the presentation of the summary results, we will follow suggestions from the instructors as well as the team managers in developing effective ways to show the summary information.

7. ACKNOWLEDGMENTS

8. REFERENCES

[1] Ben-Zvi, Dani (2007) Using Wiki to Promote Collaborative Learning in Statistics Education. In Technology Innovations in Statistics Education, vol 1, issue 1, article 4, 2007.

[2] Blei, D.M. and Ng, A.Y. and Jordan, M.I. Latent dirichlet allocation (2003). The Journal ofMachine Learning Research. Vol 3, 993—1022.

[3] Carroll, J., Neale, D., Isenhour, P., Rosson, M., Scott & McCrickard, D. (2003). Notification and awareness: synchronizing task-oriented collaborative activity. Int. J. Hum. Comput. Stud. 58(5), 605–632.

[4] Chen, H.L., Cannon, D.M., Gabrio, J., &Leifer, L. (2005, June). Using Wikis and Weblogs to Support Reflective Learning in an Introductory Engineering Design Course. Paper presented at the 2005 American Society for Engineering Education Annual Conference & Exposition.

[5] Cohen, J., (1960). A coefficient of agreement for nominal scales. Educational and Psychological Measurement, 20, 37-46.

[6] Engeström, Y. (1993). Developmental studies of work as a testbench of activity theory. In S. Chaiklin, S. and J. Lave, eds., Understanding Practice: Perspectives on Activity and Context (pp. 64–103). Cambridge: Cambridge University Press.

[7] Ganapathy, C., Shaw, E. & Kim, J. (2011) Assessing Collaborative Undergraduate Student Wikis and SVN with Technology-based Instrumentation: Relating Participation Patterns to Learning, Proceedings of the American Society of Engineering Education Conference, 2011.

[8] Glassman, M. & Kang, M. J. (2011) The logic of wikis: The possibilities of the Web 2.0 classroom. Computer-Supported Collaborative Learning (2011) 6:93–112.

[9] Kaptelinin, V., &Nardi, B. A. (2006). Acting with technology: Activity theory and interaction design. Cambridge: MIT.

[10] Moodle. http://www.moodle.org/

[11] Ma, J., Shaw, E. and Kim, J. (2011) Workflow-Based Assessment of Student Online Activities with Topic and Dialogue Role Classification, Proceedings of the AI in Education Conference, 2011.

[12] Ma, J., Shaw, E. and Kim, J. (2010) Computational Workflows for Assessing Student Learning, Proceedings of the International Conference on Intelligent Tutoring Systems Conference, 2010.

[13] Ramage, D. and Hall, D. and Nallapati, R. and Manning, C.D. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing.

[14] Rick, J. & Guzdial, M. (2006) Situating CoWeb: a scholarship of application, Computer-Supported Collaborative Learning (2006) 1:89-115.

[15] Suthers, D., Dwyer, N. and Medina, R. (2010) A framework for conceptualizing, representing, and analyzing distributed interaction Vatrapu Computer-Supported Collaborative Learning (2010) 5:5–42

[16] Upton, K. and Kay, J. (2009). Proceedings of the User Modeling, Adaptation and Personalization Conference, Narcissus: Group and Individual Models to Support Small Group Work, UMAP 2010.

[17] Wang, H.-C., Lu, C.H., Yang, J.-Y., Hu, H.-W., Chious, G.-F., Chiang, Y.-T. & Hsu, W.L. (2005) An Empirical Exploration of Using Wiki in an English as a Second Language Course, In Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies.

[18] Zimmerman, A., Lorenz, A. & Oppermann, R. (2007). An operational definition of context. In: CONTEXT 2007, Lecture Notes in Artificial Intelligence n. 4635, pp. 558–6571. Springer-Verlag, Berlin.