

# Encouraging Reading and Collaboration Using Classroom Salon

Ananda Gunawardena  
Computer Science Department  
Carnegie Mellon University  
5000, Forbes Avenue  
Pittsburgh, PA 15213  
412-268-1559  
guna@cs.cmu.edu

Aaron Tan  
School of Computer Science  
Carnegie Mellon University  
5000, Forbes Avenue  
Pittsburgh, PA 15213  
412-915-2325  
awtan@andrew.cmu.edu

David Kaufer  
English Department  
Carnegie Mellon University  
5000, Forbes Avenue  
Pittsburgh, PA 15213  
412-268-1074  
kaufer@andrew.cmu.edu

## ABSTRACT

Classroom Salon (CLS) is a special web based platform developed to encourage active reading and annotation activities. Using special highlighting and commenting tools in CLS, students indicate their understanding of concepts. CLS groups student annotations and provides a number of visualizations that can be used to understand group agreements and disagreements and other patterns of social annotations. In this paper we present data from a pilot project conducted in a second semester data structures course. We found that there is strong correlation between student's ability to find errors in a document and their overall performance in class. We also found that the majority of students prefer reading others annotations and using them to further refine their comments and hence encouraging collaborative document reading activities.

## Categories and Subject Descriptors

K.3.2 [Computer and Information Science Education]: Computer Science Education, Curriculum; H.5.3 [Information Interfaces and Presentation]: Group and Organization Interfaces; web-based interaction

## General Terms

Measurement, Design, Experimentation, Human Factors

## Keywords

Reading comprehension, annotations, data structures

## 1. INTRODUCTION

Many researchers in the behavioral sciences have studied processes of reading and comprehension. Some interested specifically in instruction have suggested reading strategies [1] as a way to improve the comprehension of the technical material in computer science courses. Reading strategies, it is thought, provide carefully crafted guidelines for the reading, interpretation and comprehension of technical material and are generally considered to be normative. However, there is no easy way to measure if students actually follow these norms in their courses. Reading as an educational activity has not been emphasized in computer science courses. Nonetheless, there have been some attempts to encourage active reading in technical courses [2] and

in particular computer science courses [3]. One approach was to "compare" the student annotations of a textbook to expert annotations using a comparison algorithm based on semantic maps. Students annotated and submitted a markup (a collection of highlights and associated comments) using the Adaptive Book software [8]. The markup elements were then extracted and analyzed using a Markup Analysis Tool [9] to give a score indicating the degree of similarity between the student markup and the expert markup.

Furthermore, there has been a growing recognition in the literature that reading is an essential component of computer science education. Chmura [1] identified reading comprehension as a key factor in determining students' success in computer science. Chmura describes a classroom culture in which students all too often jump into programming activities before taking the time to fully understand important concepts. Other authors have established a correlation between students' ability to identify key concepts in their course readings to their grades in computer science courses [2][3]. Similarly, Wileman et al [4] found a significant correlation between reading comprehension and success in computer science classes.

Despite growing understanding of the importance of reading to computer science education, the message is a hard one to convey to students. Part of the problem of reading in computer science may be that students perceive reading as a passive activity that contrasts with their preferences for active, hands-on learning. However, reading and comprehension is an important skill for CS majors who must often communicate ideas and understand requirements to work with colleagues who may be non-technical. It is also true that expert readers are active readers who do not just passively absorb reading material, but actively engage with it by asking questions and seeking ways to apply what they have read. The well-known philosopher and educator, Mortimer J. Adler [10], discusses the need to educate students to be active, critical readers who do not just regurgitate texts, but interrogate them. Adler emphasizes how active reading is a skill that must be learned and suggests that active reading may be a first step in improving overall communicative competence.

Reading, for the sake of reading may have little benefit to comprehension. Passive reading of text is often boring and

students typically ignore any instructions given with generic reading assignments. Active reading on the other hand requires students to be engaged with the text or what we call “having a conversation with the text”.

Theories of active reading, however, are hard to convey to students using static text. Can technology be fashioned to encourage the active reading of technical material in computer science courses? Despite the acknowledgment of the importance of reading among some computer science educators, there exist no tools, processes or mechanisms to integrate reading easily into computer science (CS) courses. Moreover, there are no tools to visualize the aggregated results in a way that makes sense to a computer science teacher or a student.

In this paper we describe a pilot study where we encouraged active reading and tested student’s ability to detect inaccurate or incomplete statements in a document and how that ability may or may not contribute to their overall course performance. Furthermore students were encouraged to first annotate the document individually, and then with the help of other student annotations refine their answers. The pilot subjects for this study were drawn from a second semester data structures course. We hypothesized that if a student had gained the mastery of the concepts, then he would be able to detect inaccurate or incomplete statements that are hidden among true facts in a document. The document is typically a summary of the concepts that were discussed during lectures and some errors or incomplete facts were intentionally embedded in the document. Sample documents that were used in the pilot project are given in [11].

## 2. THE METHODS AND TOOLS

### 2.1 Classroom Salon

Classroom Salon [7] (CLS) was developed at Carnegie Mellon University as part of a collaborative project between humanities, computer science and information system departments. The reading activities used in this project were performed using the CLS environment, which allows instructors to develop and upload documents. Students perform active reading in “participate” mode [Figure 1] and view how they and others have annotated the document in view mode [Figure 2]. In the view mode, students can also see the hot spots [Figure 3] where most annotations activities have taken place. Participation and view modes allows students to continue the refinement of their highlights and comments based on group annotations leading to better understanding of the things they read. Moreover, students can see, usually at the end of the assignment, where the instructor has annotated and commented and then compare their annotations to that of an expert. Students also can indicate their confidence level when making an individual annotation or when responding to instructor-generated questions using a slider (negative to positive). The slider results can be aggregated to obtain an overall confidence level of an individual student annotator or a group of

student annotators. There are many other tools in Classroom Salon, such as bar graphs indicating the confidence rankings, pie charts showing average confidence scores and word clouds showing the words most frequently used by the annotators.

CLS is an easy to use platform for assigning reading and for understanding individual and group comments on a document. Many types of reading assignments can be integrated into CLS, and in particular assignments that requires students to perform active readings using highlights and annotations to explain what they have read. In the pilot study reported here, the active reading assignments required the students to detect places in the document where incorrect or incomplete facts were hidden. As students read the document, they were expected to highlight sentences which they found to be incomplete or inaccurate. With each highlight, students were expected to include a comment to indicate why they highlighted the section [Figure 1]. Multiple highlights can be associated with a single comment to indicate the association of a comment to multiple places in the text.

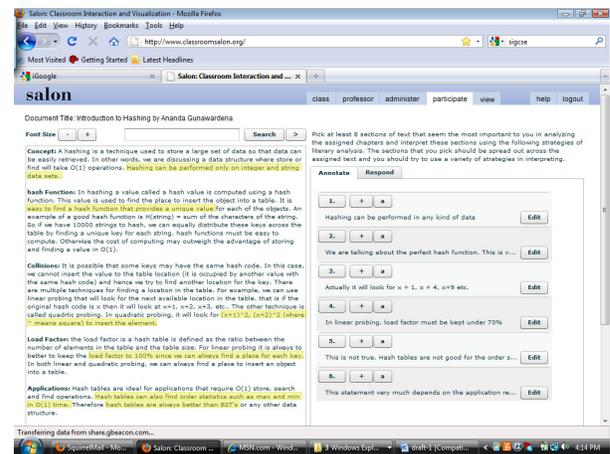


Figure 1 – Student annotations in participate mode

Once a student completed the annotation activity, he could view how others had detected common errors in the document by visualizing a summarized view of the annotations [Figure 2]. The summary presents a grid where most activity took place and individuals and their annotations grouped by paragraphs (on the right). Each horizontal row cell on the right window shows the annotations done by a user on a particular paragraph. CLS allows paragraphs to be combined to get a coarser view of the annotations.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGCSE'10, March 10–13, 2010, Milwaukee, WI, USA.  
Copyright 2010 ACM 1-58113-000-0/00/0004...\$5.00.



Figure 2 – view all annotations in a grid

## 2.2 Detecting Errors in a Document

Proof reading a paper or book involves error detection. Reviewers are expected to pay careful attention to the text content to assure that all statements in the paper or book are true and valid and even more basically, grammatical and comprehensible, before publication. In the proof-reading process we make the assumption that reviewers are subject experts and are therefore able to detect incomplete or inaccurate facts in the document. This process helps produce a high quality paper or book that is consumed by masses. We used a similar approach in this project, both to encourage active reading, and to inspect student’s ability to find errors in a document. Our hypothesis was that students who are able to correctly “review” the document have gained a level of mastery of the subject matter.

Many years of our collective experience in technical courses show that students don’t like to read. Often students perform passive reading activities with limited benefits. On the other hand many of the errors that are made during tests and assignments can be attributed to poor reading and comprehension skills. For example, in C programming, not understanding that pointers cannot be dereferenced without memory being allocated can lead to serious errors in the code. Yet a student who is able to detect this error on paper is more likely to avoid this type of debugging error during the program development stage. Our method required the instructor or subject expert to summarize the concepts covered during lectures and introduce some incomplete or inaccurate statements into the document before being published in CLS. For example, one might say that “hash tables are also ideal data structures for finding order statistics.” This incorrect fact may be hidden among some true facts such as “hash tables are efficient data structures for insert and find operations”. Experienced instructors in computer science can construct many instances of such student misconceptions. In this model, a document is prepared using many true facts and few incorrect facts hidden among the true facts. These types of documents can be prepared with little effort. All that is required of the instructor is to take a correct summary document dealing with some concept of interest. The instructor needs to think carefully about where students may misinterpret facts. The instructor then needs to deliberately alter or rewrite the facts to include incomplete or inaccurate statements.

If a student detects these errors during active reading, then we can be reasonably sure that the student has a good mastery of the subject matter. As any subject expert who carefully reads the document can detect such errors, we will be able to find out if the student is able to do the same. Moreover, since students can view a summary of all other annotations after they submit their annotations, we can also observe if some students change their annotations after seeing their peer’s annotations. We can also track how those changes can affect student learning.

## 2.3 The Pilot Project

The data and results presented in this paper are the result of two summer 2009 courses offered at the Carnegie Mellon Qatar Campus and the Carnegie Mellon Pittsburgh Campus respectively. The same standard introductory course in data structures was offered to students in each campus. There were 17 students taking the course in Qatar and 18 students taking the course in Pittsburgh. The course covers standard data structures and algorithms related to linked lists, binary search trees, heaps, hash tables, and graphs. Moreover the course also covers algorithm complexity and application efficiency in memory and speed. Students were told that 5% of the course grade is going to be on reading activities. Students were also told that as they read the documents, they must carefully identify places where facts may be incorrect or incomplete. Finally, they were told to highlight, comment and indicate their confidence level in their answers. Each course included four reading assignments covering the concepts of linked lists, binary search trees, hashing and heaps. Students were asked to complete the reading assignments after the concepts were covered in class. Students were typically given 2-3 days to complete the reading and annotation assignment. Students became reviewers of the concept document that the instructor has provided. Classroom Salon requires that students justify a highlight by providing a corresponding annotation to explain why that part of the text was highlighted. This allows us to confirm why a student highlighted a particular section, their justification for their action. In CLS, a comment can include multiple highlights to allow association of a single comment to multiple places in the text.

## 2.4 Annotation Visualization

The novelty of the Classroom Salon platform is its ability to aggregate all comments into a grid so all comments can be analyzed as a group. This allows us to understand, using annotation visualization, students’ specific thoughts related to a particular section of the document as well as whether their thoughts converge or diverge. There are many levels of annotation visualization in CLS. In level 1, student can select the document to see the aggregated annotations. As shown in Figure 2, students can see a grid representing the annotations. In level 2, students can just focus on annotations from a few students and understand how their annotations compared to that of their selected group. CLS provides more inquiry into annotated sections by changing parameters such as the number of common word overlaps, how many words define a hot spot and how much similarity tolerance is applied in annotation visualization.

The grid display shown in Figure 2 indicates the annotations made on a collection of paragraphs (x-axis) and users who made the annotations (y-axis). It is easy to see which paragraphs of the text had most of the activity or the students who selected the same paragraphs as the instructor. Clicking on any grid cell highlights the text on the left corresponding to the comment. The level 3

probe of annotations involves finding out exactly who said what and where. For example, one can click on a hot spot (a place indicating the density of annotation activity) to see the users who were involved in that position as shown in Figure 3.



Figure 3 – Hot Spots

Specific location with a hot spot (as shown in left window of Figure 3) indicates those students who were the participants of a particular section. This feature serves two purposes. One, this allow students to see who else has annotated the same location as them and two, it also allow students to read the comments of others and see whether he agrees with other students or not. More importantly, we can detect if the student is willing to change his responses based on what he think is the majority view. In order to allow students to freely express their views, students were not penalized for giving a wrong answer or interpretation. This allowed us to find out how a student response to a question might be influenced by his peers and who among the students might be having trouble interpreting an important concept.

### 3. RESULTS

All annotation data (specific content, student ID, assignment ID, location of the annotation, student confidence of the annotation) were extracted from the CLS database and analyzed to understand how students' ability to detect errors vary from assignment to assignment. For each assignment, we needed to understand how many students actually identified the positions where incomplete or inaccurate information was planted. This information was compared to the expert annotation information. The locations did not have to coincide exactly with the expert and a small margin of error was acceptable. For example, the correct highlight may involve specific words in the text as indicated by the expert, but students may have selected not all, but a few of them. We extracted this data from the database for each of the concepts highlighted in the same textual range. We plotted the following graphs as shown by Figures 4 and 5. Figure 5 shows the number of students who were able to find the errors in each of the locations for the binary search tree assignment. The x-axis indicates the number of errors (max 7) and y-axis indicate the number of students who correctly identified those errors. For example, eight students were able to correctly identify only 1 error but no student was able detect more than 4 errors.

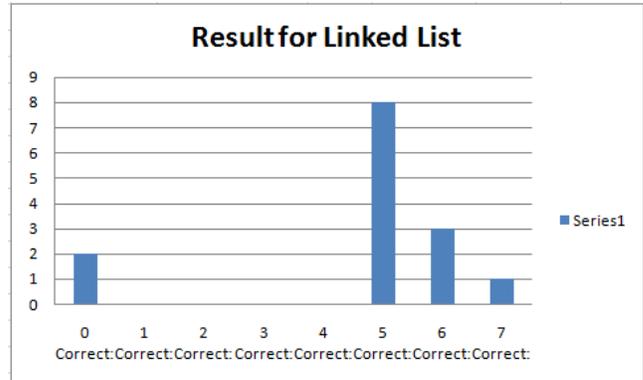


Figure 4 – Results for Linked Lists

More students succeeded in finding errors in the Linked List assignment. As shown in Figure 4, eight students identified five errors (out of 7) and one student was able to identify all the errors.

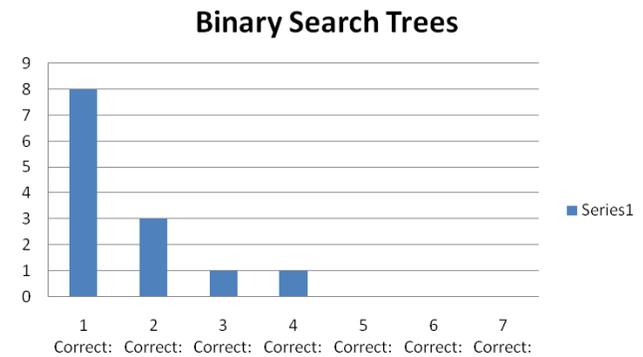


Figure 5 – BST Assignment

The other two assignments (heaps and hash tables) had results similar to BST assignment. It was evident that students had trouble identifying more errors as the complexity of the concept increased. We also compared the number of students who correctly identified specific errors. We calculated the average student correctness, on a scale of 0 to 1, where 1 is the highest possible error detection rate (equivalent to an expert). The graph [Figure 6] seems to suggest a correlation between student's ability to detect errors and their grade in the course.

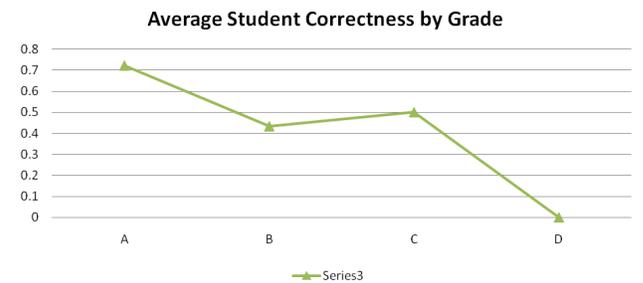


Figure 6 – Average Correctness

We also surveyed students to understand their perception towards reading in the data structures course and if they thought they learned from the annotations of others. Survey results revealed

some interesting observations. Figure 7 shows that most students spent between 15 and 60 minutes per assignment working on detecting error activities.

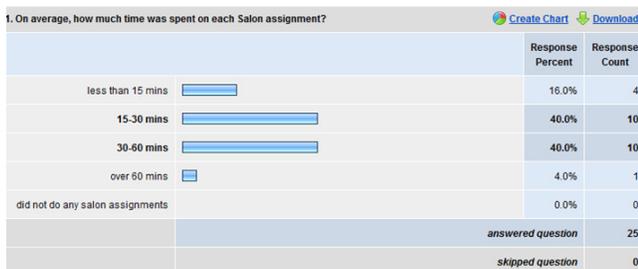


Figure 7 – Time Spent on Reading

92% of the students responded that they looked at other students’ annotations once they completed theirs. 72% of the students sometimes changed their annotations as a result of seeing what others had done while 16% never did change their annotations. We note that CLS encouraged students to rethink their annotations after seeing what others had annotated. For some assignments we asked students to respond to global questions, questions that were raised in the context of the entire document, not just on a particular section. We asked students to indicate their level of confidence that their answer was correct. 56% of the students said they thought about assigning this confidence level carefully. Although we did not analyze how the accuracy of the answers relate to students perception in this study, it would be interesting to analyze this data in the future. Overall, 25 students who responded to the survey had positive comments about the Classroom Salon platform and the reading process. All comments are available from [11].

#### 4. CONCLUSION

The pilot project described here was designed to encourage more reading and annotation activities in early computer science courses where reading is important but undervalued. The preliminary results of the pilot indicate that there may be a stable correlation between student’s ability to successfully review a document and their course performance. Although sample size is small, we plan to try to replicate these findings for a large set of data. It seems intuitive that those students who read and understand documents are likely to score better in the labs and tests. The Classroom Salon platform is also a new and interesting conversation platform. Unlike discussions boards, where people make and read comments outside of the context, in the CLS model, all commenting and annotations are done in the context of the document and hence are more likely to be effective. Furthermore, students seem to learn from others annotations, and in the process either change their responses or argue against other responses. Another use of CLS may be for code review activities where students can be encouraged to carefully think about their

code before engaging in actual coding activities. CLS can also be used to determine if the ability to detect and explain incorrect or incomplete concepts on paper actually contributes to the reduction of errors in code and hence reduce the debugging time. Classroom Salon is freely available [7] to any instructor wishing to implement active reading activities in their courses.

#### 5. ACKNOWLEDGMENTS

We would like to thank Joanna Wolfe from the University of Louisville and Alex Cheeks from CMU-Qatar for their role in developing Classroom Salon. Also we thank Raja Sooriaymoorti from CMU’s Information Systems department and the original development group from CMU’s Information Systems department for their tireless effort to make CLS a novel annotation and visualization platform.

#### 6. REFERENCES

- [1] Chmura, G., 1999. Using Reading Strategies to Help Computer Science Students Better Understand Technical Material. [http://gse.gmu.edu/research/tr/tr\\_languagearts](http://gse.gmu.edu/research/tr/tr_languagearts)
- [2] Gunawardena, A. 2008 "Work in progress: Measuring reading comprehension in technical courses." Proceedings of the Conference on Frontiers in Education (FIE), Albany, NY.
- [3] Gunawardena, A, Barr, J, Owens A 2008. , A method for analyzing reading comprehension in computer science courses, Proceedings of the 13th annual conference on Innovation and technology in computer science education, Madrid, Spain, Pages 348-348 ISBN:978-1-60558-078-4
- [4] Wileman S., Konvalina, J, Stephens, L. 1981. Factors Influencing Success in Beginning Computer Science Courses. Journal of Educational Research, Vol. 74, 1981
- [5] Wade, S.E., & Moje, E. B. 2000. The role of text in classroom learning. In M.L. Kamil, P.B. Mosenthal, P.D. Pearson, & R. Barr (Eds.), *Handbook of reading research: Volume III* (pp. 609-627). Mahwah, NJ: Erlbaum.
- [6] Cheeks A. 2009. Classroom Salon, An interactive Platform for annotation and visualization, Masters Thesis, Communications Design Department, Carnegie Mellon University
- [7] Classroom Salon, 2009 – An interactive Platform for Collaboration and Visualization, <http://www.classroomsalon.org>
- [8] Adaptive Book Project, 2004, Carnegie Mellon University, <http://www.cs.cmu.edu/~ab>
- [9] Markup Analysis Tool, 2007 Carnegie Mellon University, Patent Pending. <http://srv01.pragma.cs.cmu.edu:1600/>
- [10] Adler, M. J. 1940,1967. How to read a book: The art of getting a liberal education. New York: Simon and Schuster.
- [11] A Repository of Documents for Implementing the Detection Model, 2009, <http://www.cs.cmu.edu/~ab/DMR>