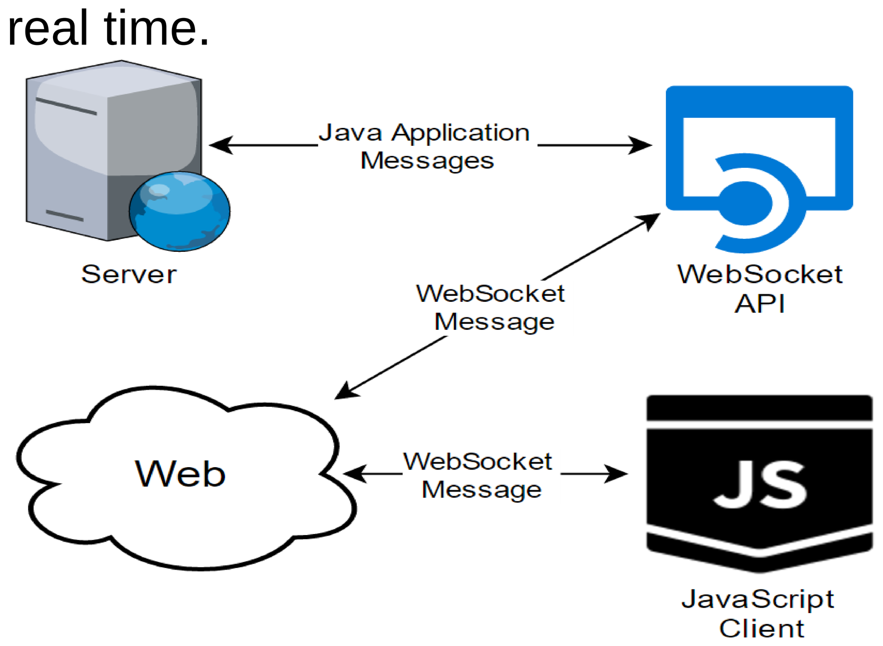
COMPS: Updating and Improving a Web-based Application for Collaborative Educational Computer-Monitored Problem-Solving Discussions

John Carden   Advisors: Drs. Jung Hee Kim, Michael Glass, and Kelvin Bryant
Department of Computer Science
North Carolina A&T State University

Introduction

COMPS (Computer-Mediated Problem Solving) is web-delivered chat application employed at NC A&T for small-group problem-solving. Students collaborate to answer questions in Computer Science classes at A&T, focusing on learning and applying Java concepts. COMPS chat interaction logs have been used in studies of text analytics and the dynamics of collaborative learning.

The COMPS server runs Java while the client uses JavaScript, both communicating through WebSockets. WebSockets is an advanced API which permits highly interactive communication between a server and browser, the students can see and respond to each other's keystrokes in real time.



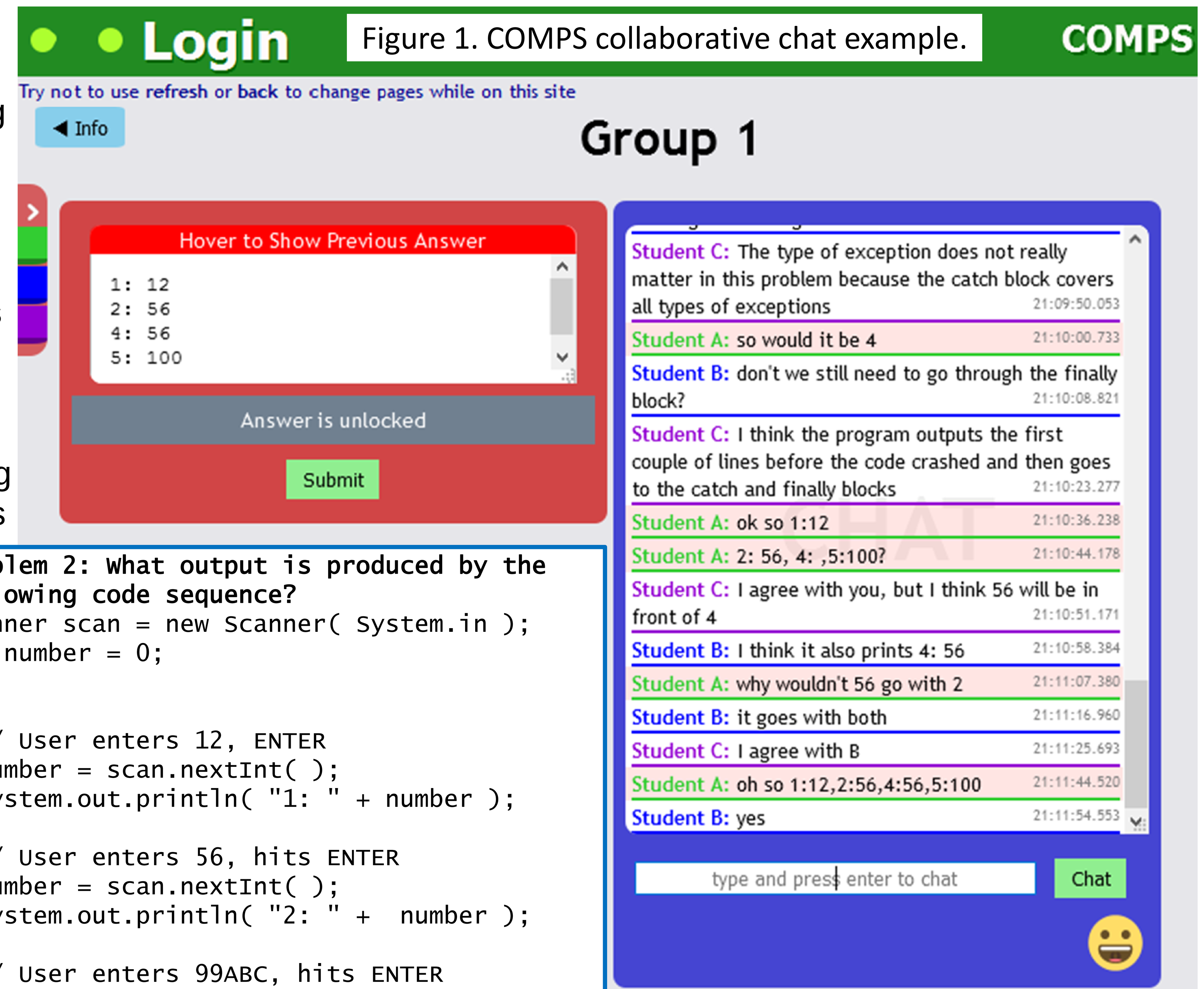
My role is to develop and maintain the application and work with the website team to facilitate communication between the COMPS website and the COMPS chat application.

Example Session

Figure 1 shows an exchange between 3 students who are solving problem 2 shown below and to the right. You can see the students collaborative problem solving skills in action as they work to find the problem's solution. Interactions like these further the understanding we have of how students learn and how we can improve that process.

In pursuit of improving the accuracy of our research, I have updated the application to allow for the IP addresses to be logged in every message. This permits messages to be connected to survey results directly and improves the accuracy of the analytics research our team performs.

Figure 1. COMPS collaborative chat example.




```
Problem 2: What output is produced by the
following code sequence?
Scanner scan = new Scanner( System.in );
int number = 0;
try
{
  // User enters 12, ENTER
  number = scan.nextInt( );
  System.out.println( "1: " + number );

  // User enters 56, hits ENTER
  number = scan.nextInt( );
  System.out.println( "2: " +  number );

  // User enters 99ABC, hits ENTER
  number = scan.nextInt( );
  System.out.println( "3: " +  number );
}
catch( Exception e)
{
  String s = scan.nextLine( );
  System.out.println( "4: " + number );
}
finally
{ // User enters 100, hits ENTER
  number = scan.nextInt( );
  System.out.println( "5: " + number );
}
Answer (4 lines):
1: 12   2: 56   4: 56    5: 100
```

Using the application allows the students to reach an agreement, then validate their answer with the teaching assistant. They can then repeat the process for each question.

Following this method engages all students, prompts them to articulate their reasoning, and reduces shallow reasoning and guessing.

Learning Result

We analyzed learning gains according to the relative position of each student in a discussion group (highest, middle, lowest), based on pre-test score. The high student in each discussion lost ground on average, the other two gained.

Table 1. Average Learning Gains

	Highest Stratum	Middle	Lowest	Class average
Lab3 (n=32)	-0.32	0.45	0.34	0.23
Lab4 (n=24)	0.02	0.34	0.47	0.32

However the post-test scores for high, mid, and low students tended to become more alike.

Table 2. Lab 3 Pre-test and Post-test Scores

	Pre-Test Mean / SD		Post- Test Mean / SD		t	Sig
High	12.25	3.41	9.75	6.06	t (11)= 1.94	p =.078
Mid	5.92	5.05	12.25	4.41	t (11)= -3.74	p =.003
Low	3.75	3.1	9.25	4.56	t (7)= -4.72	p = .002

Students rated the last two labs more highly than the first two labs, possibly as they became more accustomed to the group exercises.

Table 3: After lab surveys

	Effectiveness of group work Mean / SD		Understanding of concept Mean / SD		Interest in lab Mean /SD	
Lab1	3.17	0.68	3.45	0.96	3.19	0.94
Lab2	3.08	0.93	3.42	1.05	3.08	0.93
Lab3	3.47	0.71	4.03	1.06	3.65	0.76
Lab4	3.40	0.61	3.78	0.85	3.17	0.89

COMPS Chat Timing Phenomena

COMPS chat differs from normal conversation. Figure 2 shows dialogue between three students, where B and C both responded to A nearly simultaneously. Figure 3 shows the timing of the keystroke messages. The black lines represent each keystroke, lower grey lines represent backspace-deletions, and the red lines represent finalizing the chat message by pressing enter.

- In regular conversation student B, who had the more substantive thought, would have had to wait for C to finish speaking. Instead B can start right away.
- There is a three second gap before student B's and student C's responses. In regular conversation three seconds can be an unusually long and awkward pause. In our dialogues people seem willing to let each other have think time.
- B's answer took a long time, containing pauses and corrections. The other students can witness B's typing in real time, following B's thought process as the statement is revised.

A	I think its B D and E ←	
B		I agree with B,D, and E ←
C		Actually no im changing to just E←

Figure 2. Example of simultaneous response.

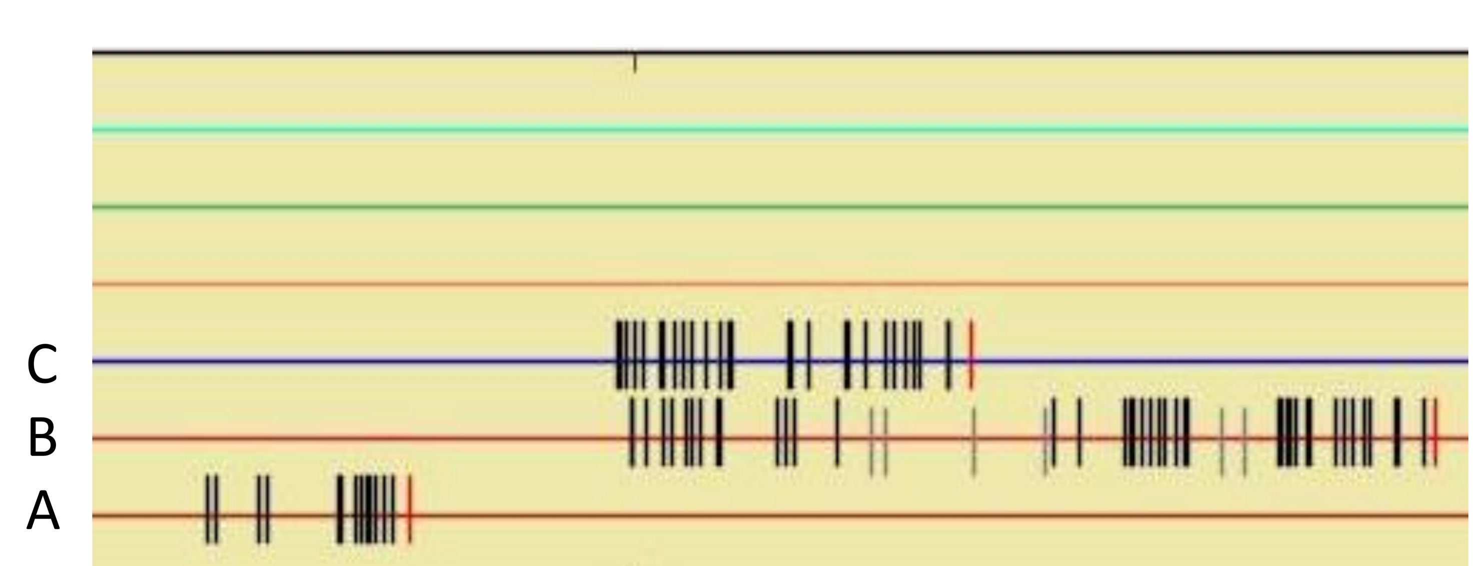


Figure 3. Keystroke timing diagram.

Future Work

Other people in the COMPS project are working on text analytics to produce measures of conversation quality, e.g. how much students are participating and how often they show evidence of understanding. We plan to add an instructor dashboard to the COMPS server, using text analysis to monitor the conversations in real time.

Acknowledgement

Partial support for this work was provided by the National Science Foundation's Improving Undergraduate STEM Education (IUSE) program under Award No. 1504917. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.