Cross Talk, Students Typing Simultaneously in Typed Chat
Chinedu Emeka[1,2], Yesukhei Jagvaral[1,3], Brian Sulikowicz[1]

Introduction

When students type-chat all at the same time, how is that different from normal conversation? Can they really read, think, and respond in real time?

This project studies computer-mediated small-group collaborative discussions of CS Java programming problems. Students chat via typing. Each chat group is typically made up of 3 or 4 people, an instructor or TA oversees each group.

An unusual feature is that students can type and see each other’s typing simultaneously. They do so quite a bit. Earlier studies show that the students engage in productive conversation, even when they are chatting simultaneously.

Here we looked at a collection of simultaneous chat incidents, categorizing them according to where was the antecedent—the utterance being responded to.

We have also started some numerical analyses of the chat behavior, discovering how long a person has to pause before other people consider normal turn-taking behavior, and how they are typing differently in the simultaneous regime.

Conclusions, Future Work

Simultaneous chatting discourse behaviors have not been well-studied. In this school year we have shown:

- Students are making use of the facility about 15% of the time.
- When they don’t take turns, the simultaneous regime, students are not reading, thinking, and typing at the same time. They are responding to an earlier statement, as in normal dialogue.
- In the simultaneous regime, we see three patterns of overlapped conversation.
- Even though they don’t process the other person’s typing in real time, there are observable differences in behavior when another person is typing.

Current and future work concentrates on:

A. the pauses and gaps in typing, fitting a gamma-like distribution and explaining the differences
B. trying to reliably categorize and count the different response behaviors in a large sample of text.

Example Dialogue

Who
Dialogue Turn
A: 99ABC isn’t compatible with an int type, so what type of exception would that be
B: InputMismatchException
B: or NumberFormatException?
A: wouldn’t it be numberFormatException because I thought that InputMismatch was when let’s say you’re checking for more things than there are in the file
B: what do you mean more things than there are in the file?
B: Input mismatch is if you’re supposed to be taking in an int like using scan.nextInt but you’re reading in a string instead
C: The type of exception does not really matter in this problem because the catch block covers all types of exceptions
A: so would it be 4
A: 2: 56, 4: .5100?
C: I agree with you, but I think 56 will be in front of 4
B: I think it also prints 4: 56
B: it goes with both
C: I agree with B
C: oh so 1:12,2:56,4:56,5:100

B: yes

Acknowledgements

Partial support for this work was provided by the National Science Foundation’s Improving Undergraduate STEM Education (IUSE) program under Award No. 1024917. 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. This work has been in collaboration with professors and students at North Carolina A&T State University, where the project is based.

VU faculty adviser is Michael Glass.