Toward Conversational Agents that Support Learning: A Look at Human Collaborations in Computer Science Problem Solving

Fernando J. Rodríguez
University of Florida
Gainesville, FL 32601, USA
fjrodriguez@ufl.edu

Kimberly Michelle Price
University of Florida
Gainesville, FL 32601, USA
kimberlymprice@ufl.edu

Mickey Vellukunnel
University of Florida
Gainesville, FL 32601, USA
m.vellukunnel@ufl.edu

Kristy Elizabeth Boyer
University of Florida
Gainesville, FL 32601, USA
keboyer@ufl.edu

Every submission will be assigned their own unique DOI string to be included here.

Abstract
As conversational systems become more common, users will start to demand more sophisticated functionality from them. Using these systems to support learning is one such function, as many users already view their systems as sources of knowledge. In order to support users in learning tasks and problem solving, we need to gain a better understanding of how collaborative conversations unfold in these domains. This paper presents a set of phenomena observed during pair programming studies with undergraduate students, focusing on how the conversations are structured. These ideas can help inform the design of conversational systems that adapt to specific situations that occur during learning, fostering natural and effective conversations.

Author Keywords
Collaborative conversation; learning support; textual dialogue

Introduction
Conversational agents are being used increasingly in people's daily interactions. We ask them the weather, tell them to remind us of our schedule, and use them to help us search the web on our computers and phones. People see these systems as personal assistants, and as these systems grow in popularity, users start demanding more

of them. In certain promotional material, users are shown asking conversational agents factual information, such as the distance between our planet and the moon. These systems have great potential to serve as educational agents and even to become partners who actively engage humans in learning.

In order to understand how people use conversation in learning activities, we need to investigate how conversations unfold in learning environments. In many learning environments today, collaboration is a central practice, so understanding how peers engage in conversation during learning holds great promise for informing the design of conversational agents that support learning. Our research group has studied collaborative dialogue in numerous contexts including spoken in-person conversation, remote computer-mediated synchronous textual conversation [1], and textual asynchronous discussions [4]. In this position paper we discuss explorations of how textual conversations are structured in computer science learning with undergraduate students. We focus on textual dialogue because it is a comfortable form of communication for "digital natives" and holds other benefits such as an accurate transcript that the user can refer back to during the learning process.

Textual Conversations during Programming

A large portion of our work has focused on how students engage in conversations around computer science problem solving, specifically programming. We conducted a study in which undergraduate students worked in pairs to complete a programming task using a block-based programming language called *Snap!* [3]. The students followed roles according to the pair programming paradigm from software engineering, which has been shown highly effective in supporting programming for learners [2]: the *driver* creates and edits the program, and the *navigator* provides instructions and feedback [5]. Students collaborated remotely, with the driver sharing her screen with the navigator and communicating through textual chat. After each one-hour session, we exported the chat logs of the pairs along with all of the coding actions. We manually labeled each utterance with a dialogue act tag, and an inter-annotator agreement study showed strong agreement (κ=0.74).

The following subsections describe phenomena that we observed from our participants during the study, with particular attention on how they structured their responses to different types of utterances.

Directives vs. Suggestions

When communicating with the driver, the navigator would often ask the driver to complete some action in order to achieve the goal of the task. These requests came in two forms: *directives* (explicit) and *suggestions* (implicit). The phrases "use an if statement" and "we could use an if statement" have the same illocutionary intent, that the driver would complete some action, but their tone differs. The number of total directives (μ=15.81, σ=17.28, max=67, min=0) was statistically significantly higher than the number of total suggestions (μ=6.11, σ=4.13, max=16, min=0) (p=0.0082; independent two-sample t-test). We observed the same pattern with regard to relative frequency, adjusted for the total number of messages in a collaboration. The high variation and range in number of directives between sessions suggests that the use of suggestions versus directives varied widely across pairs, with some individuals tending to communicate through suggestions while others preferred directives. Most would use a combination of both. Importantly, we observed that when a student would begin to give directives, she often did so in small, separate sequential messages rather than in one large one.

Providing Meta-Comments
We refer to reflections on the problem-solving process (e.g., "I see what's going on"; "nevermind") as *meta-comments*. In most cases, these messages were used as a way to keep the conversation going, serving as filled pauses (e.g., "hmmm" to indicate the student is thinking). These meta-comments, particularly when given by the navigator, were significantly positively associated with learning in our recent study [3]. This insight may be particularly relevant for conversational agents which are likely to serve a "navigator" role when engaging students in dialogue around learning.

Driver: Which choice do I click for the arrow in it?

Nav: button clicked i think

Nav: are the operations on the right below the while screen actual buttons?

Driver: Oh.

Driver: I didn't even notice that.

Driver: They are

Excerpt 1: Question and answer interaction

Driver: what is an operand...?

Nav: like + - divide

Driver: ooh ok gotcha

Nav: and they fill in an operand

Nav: if its right we print right

Nav: if they put like + which is wrong we say wrong

Excerpt 2: Answer and explanation interaction

Expressing and Resolving Uncertainty
We also examined the ways in which students expressed *uncertainty* as explicit statements of confusion (e.g., "I don't know") and hedged suggestions (e.g., "I think it's because..."). We scored task quality according to adherence to the requirements, and found that students with high-quality solutions were able to successfully resolve uncertainty when it emerged, while low-scoring pairs would often ignore it or have multiple unsolved uncertainty events at a time. This finding suggests that resolving uncertainty quickly during problem solving may foster improved learning.

Conversation and Task Actions
During the pair programming collaborations we studied, students' main form of communication with each other was textual chat. However, we also noticed that many times, drivers "responded" to their partner's utterance by taking a programming action, not by typing a dialogue message. Similarly, a textual chat message can serve as a response to a programming action in the form of a *feedback* message from either driver or navigator. We found that feedback from both students in a pair was significantly more frequent in pairs that achieved a higher score in the programming task than those that achieved medium or low scores [3]. A conversational agent must model its users' actions as well as their utterances in order to be most effective.

Question/Answer Dynamics
Questions and answers between students in a pair may be especially relevant for designing conversational agents. Excerpt 1 shows an interaction in which the driver asks a question, which is answered by the navigator and followed by a navigator question. It is worth noting that the two driver messages before her answer are possibly a result of continuing the conversation without realizing that a question had been asked. This phenomenon is recognized in textual human-human conversations, in which a speaker may author a message even as her partner is already sending a different utterance.
We noticed once again that our users tended to split their utterances into multiple messages, particularly for lengthy answers to a question that had been posed. When a student asked a question in these cases, the partner's main answer would be her next message, but then she would elaborate on the answer further in subsequent messages (see Excerpt 2). This approach may serve a validation role by providing reassurance that the given answer was appropriate. In textual conversation, it is also common for a person to send multiple short messages to ensure her recipient that she is actively participating in the conversation at the moment. Perhaps most importantly, providing many short messages helps to avoid overwhelming the reader with one long message, and should be considered whenever providing support via textual conversation. In a learning context, it may be useful for an agent to give a response and wait to see if the user understood. If no response is given, the agent might elaborate on the answer to continue the conversation. These design choices may foster a sense of the agent as an active participant in a conversation.

Dialogue with a "Knowledgeable Collective"

In addition to the synchronous dialogue we have previously discussed, another one of our research directions involves asynchronous textual discussions in support of classes. Although these discussions differ in many ways from synchronous dialogues, we believe that they may hold important insights for one reason in particular: when users ask questions of many state-of-the-art virtual assistants today, those virtual assistants are querying multiple resources constructed by many different people for many different purposes in order to find the best answer. In this way, the user is actually asking a question of a knowledgeable collective. There are times, and may increasingly be times, that explicitly acknowledging that dynamic could lead to more effective conversations. In asynchronous discussion boards, students are posing questions to the knowledgeable collective (their peers, teaching assistants, and classmates). These messages, and the responses to them, could lead to insight on the design of some aspects of conversational agents. On the discussions we have analyzed, nearly all begin with a student asking a question about homework or a project. This is usually followed by students' (36%) and/or instructors' answer (73%), and follow-up discussions (65%) on these questions or answers. Instructors can mark students' answers as "good answers," endorsing them [4].

An important implication for conversational UX design to support learning is the notion of putting users under less "pressure" than they feel when seeking help from instructors or teaching assistants. Discussion board users report lower sense of pressure, and they also hold the benefit of availability which conversational agents can also provide.

Conclusion

Conversational agents capable of holding a natural conversation with humans are closer to becoming a reality. We can move toward sophisticated conversation design in these systems by developing an understanding of how humans structure their conversations, particularly in the important domain of supporting learning. It is important to observe how conversations in learning contexts unfold, what makes them unique, and what makes them effective.

References

[1] Aysu Ezen-Can and Kristy Elizabeth Boyer. 2015. Understanding student language: An unsupervised dialogue act classification approach. *Journal of Educational Data Mining* 7, 1 (2015), 51–78.

[2] Alex Radermacher and Gursimran S. Walia. 2011. Investigating the Effective Implementation of Pair Programming: An Empirical Investigation. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education (SIGCSE '11)*. 655–660.

[3] Fernando J. Rodríguez, Kimberly Michelle Price, and Kristy Elizabeth Boyer. 2017. Exploring the Pair Programming Process: Characteristics of Effective Collaboration. In *Proceedings of the 48th ACM Technical Symposium on Computer Science Education (SIGCSE '17)*. to appear.

[4] Mickey Vellukunnel, Philip Sheridan Buffum, Kristy Elizabeth Boyer, Jeffrey Forbes, Sarah Heckman, and Ketan Mayer-Patel. 2017. Deconstructing the Discussion Forum: Student Questions and Computer Science Learning. In *Proceedings of the 48th ACM Technical Symposium on Computer Science Education (SIGCSE '17)*. to appear.

[5] Laurie Williams. 2001. Integrating Pair Programming into a Software Development Process. In *Proceedings of the 14th Conference on Software Engineering Education and Training*. 27–36.