Impact of Initiative on Collaborative Problem Solving [*]

Cynthia Kersey
Department of Computer Science
University of Illinois at Chicago
Chicago, Illinois 60613
ckerse2@uic.edu

Abstract

Even though collaboration in peer learning has been shown to have a positive impact for students, there has been little research into collaborative peer learning dialogues. We analyze such dialogues in order to derive a model of knowledge co-construction that incorporates initiative and the balance of initiative. This model will be embedded in an artificial agent that will collaborate with students.

1 Introduction

While collaboration in dialogue has long been researched in computational linguistics (Chu-Carroll and Carberry, 1998; Constantino-González and Suthers, 2000; Jordan and Di Eugenio, 1997; Lochbaum and Sidner, 1990; Soller, 2004; Vizcaíno, 2005), there has been little research on collaboration in peer learning. However, this is an important area of study because collaboration has been shown to promote learning, potentially for all of the participants (Tin, 2003). Additionally, while there has been a focus on using natural language for intelligent tutoring systems (Evens et al., 1997; Graesser et al., 2004; VanLehn et al., 2002), peer to peer interactions are notably different from those of expert-novice pairings, especially with respect to the richness of the problem-solving deliberations and negotiations. Using natural language in collaborative learning could have a profound impact on the way in which educational applications engage students in learning.

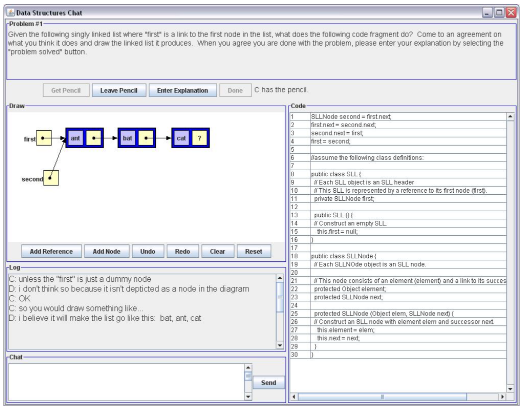
There are various theories as to why collaboration in peer learning is effective, but one of that is commonly referenced is co-construction (Hausmann et al., 2004). This theory is a derivative of constructivism which proposes that students construct an understanding of a topic by interpreting new material in the context of prior knowledge (Chi et al., 2001). Essentially, students who are active in the learning process are more successful. In a collaborative situation this suggests that all collaborators should be active participants in order to have a successful learning experience. Given the lack of research in modeling peer learning dialogues, there has been little study of what features of dialogue characterize co-construction. I hypothesize that since instances of co-construction closely resemble the concepts of control and initiative, these dialogue features can be used as identifiers of co-construction.

While there is some dispute as to the definitions of control and initiative (Jordan and Di Eugenio, 1997; Chu-Carroll and Brown, 1998), it is generally accepted that one or more threads of control pass between participants in a dialogue. Intuitively, this suggests that tracking the transfer of control can be useful in determining when co-construction is occurring. Frequent transfer of control between participants would indicate that they are working together to solve the problem and perhaps also to construct knowledge.

The ultimate goal of this research is to develop a model of co-construction that incorporates initiative and the balance of initiative. This model will be embedded in KSC-PaL, a natural language based peer agent that will collaborate with students to solve

[*] This work is funded by NSF grants 0536968 and 0536959.

Proceedings of the ACL-08: HLT Student Research Workshop (Companion Volume), pages 43–48,
Columbus, June 2008. ©2008 Association for Computational Linguistics



Given the following singly linked list where "first" is a link to the first node in the list, what does the following code fragment do? Come to an agreement on what you think it does and draw the linked list it produces. When you agree you are done with the problem, please enter your explanation by selecting the "problem solved" button.

C: unless the "first" is just a dummy node
D: i don't think so because it isn't depticted as a node in the diagram
C: OK
C: so you would draw something like...
D: i believe it will make the list go like this: bat, ant, cat

```
SLLNode second = first.next;
first.next = second.next;
second.next = first;
first = second;

//assume the following class definitions:

public class SLL {
  // Each SLL object is an SLL header
  // This SLL is represented by a reference to its first node (first).
  private SLLNode first;

  public SLL () {
  // Construct an empty SLL.
    this.first = null;
}

public class SLLNode {
  // Each SLLNOde object is an SLL node.

  // This node consists of an element (element) and a link to its succes
  protected Object element;
  protected SLLNode next;

  protected SLLNode (Object elem, SLLNode next) {
  // Construct an SLL node with element elem and successor next.
    this.element = elem;
    this.next = next;
  }
}
```

Figure 1: The data collection interface

problems in the domain of computer science data structures.

In section 2, I will describe how we collected the dialogues and the initial analysis of those dialogues. Section 3 details the on-going annotation of the corpus. Section 4 describes the future development of the computational model and artificial agent. This is followed by the conclusion in section 5.

2 Data Collection

In a current research project on peer learning, we have collected computer-mediated dialogues between pairs of students solving program comprehension and error diagnosis problems in the domain of data structures. The data structures that we are focusing on are (1) linked lists, (2) stacks and (3) binary search trees. This domain was chosen because data structures and their related algorithms are one of the core components of computer science education and a deep understanding of these topics is essential to a strong computer science foundation.

2.1 Interface

A computer mediated environment was chosen to more closely mimic the situation a student will have to face when interacting with KSC-PaL, the artificial peer agent. After observing face-to-face interactions of students solving these problems, I developed an interface consisting of four distinct areas (see Figure 1):

1. Problem display: Displays the problem description that is retrieved from a database.
2. Code display: Displays the code from the problem statement. The students are able to make changes to the code, such as crossing-out lines and inserting lines, as well as undoing these corrections.
3. Chat Area: Allows for user input and an interleaved dialogue history of both students participating in the problem solving. The history is logged for analysis.
4. Drawing area: Here users can diagram data structures to aid in the explanation of parts of the problem being solved. The drawing area has objects representing nodes and links. These objects can then be placed in the drawing area to build lists, stacks or trees depending on the type of problem being solved.

The changes made in the shared workspace (drawing and code areas) are logged and propagated to the partner's window. In order to prevent users from making changes at the same time, I implemented a system that allows only one user to draw or make changes to code at any point in time. In order to make a change in the shared workspace, a user must request the ”pencil” (Constantino-González and Suthers, 2000). If the pencil is not currently allocated to her partner, the user receives the pencil and can make changes in the workspace. Otherwise, the partner is informed, through both text and an audible alert, that his peer is requesting the pencil. The chat area, however, allows users to type at the same time, although they are notified by a red circle at the top of the screen when their partner is typing. While, this potentially results in interleaved conversations, it allows for more natural communication between the peers.

Using this interface, we collected dialogues for a total of 15 pairs where each pair was presented with five problems. Prior to the collaborative problem solving activities, the participants were individually given pre-tests and at the conclusion of the session, they were each given another test, the post-test. During problem solving the participants were seated in front of computers in separate rooms and all problem solving activity was conducted using the computer-mediated interface. The initial exercise let the users become acquainted with the interface. The

Predictor	Prob. 3 (Lists)	Prob. 4 (Stacks)	Prob. 5 (Trees)
Pre-Test	0.530 (p=0.005)	0.657 (p=0.000)	0.663 (p=0.000)
Words	0.189 (p=0.021)		
Words per Turn	0.141 (p=0.049)		
Pencil Time	0.154 (p=0.039)		
Total Turns	0.108 (p=0.088)		
Code Turns			0.136 (p=0.076)

Table 1: Post-test Score Predictors (R^2)

participants were allowed to ask questions regarding the interface and were limited to 30 minutes to solve the problem. The remaining exercises had no time limits, however the total session, including pre-test and post-test could not exceed three hours. Therefore not all pairs completed all five problems.

2.2 Initial Analysis

After the completion of data collection, I established that the interface and task were conducive to learning by conducting a paired t-test on the pre-test and post-test scores. This analysis showed that the post-test score was moderately higher than the pre-test score (t(30)=2.83; p=0.007; effect size = 0.3).

I then performed an initial analysis of the collected dialogues using linear regression analysis to identify correlations between actions of the dyads and their success at solving the problems presented to them. Besides the post-test, students solutions to the problems were scored, as well; this is what we refer to as problem solving success. The participant actions were also correlated with post-test scores and learning gains (the difference between post-test score and pre-test score). The data that was analyzed came from three of the five problems for all 15 dyads, although not all dyads attempted all three problems. Thus, I analyzed a total of 40 subdialogues. The problems that were analyzed are all error diagnosis problems, but each problem involves a different data structure - linked list, array-based stack and binary search tree. Additionally, I analyzed the relationship between initiative and post-test score, learning gain and successful problem solving. Before embarking on an exhaustive manual annotation of initiative, I chose to get a sense of whether initiative may indeed affect learning in this context by automatically tagging for initiative using an approximation of Walker and Whittaker's utterance based allocation of control rules (Walker and Whittaker, 1990). In this scheme, first each turn in the dialogue must be tagged as either: (1) an assertion, (2) a command, (3) a question or (4) a prompt (turns not expressing propositional content). This was done automatically, by marking turns that end in a question mark as questions, those that start with a verb as commands, prompts from a list of commonly used prompts (e.g. ok, yeah) and the remaining turns as assertions. Control is then allocated by using the following rules based on the turn type:

1. Assertion: Control is allocated to the speaker unless it is a response to a question.

2. Command: Control is allocated to the speaker.

3. Question: Control is allocated to the speaker, unless it is a response to a question or a command.

4. Prompt: Control is allocated to the hearer.

Since the dialogues also have a graphics component, all drawing and code change moves had control assigned to the peer drawing or making the code change.

The results of the regression analysis are summarized in tables 1 and 2, with blank cells representing non-significant correlations. Pre-test score, which represents the student's initial knowledge and/or aptitude in the area, was selected as a feature because it is important to understand the strength of the correlation between previous knowledge and post test score when identifying additional correlating features (Yap, 1979). The same holds for the time related features (pencil time and total time). The remaining correlations and trends to correlation suggest that participation is an important factor in successful collaboration. Since a student is more likely to take initiative when actively participating in prob-

Predictor	Prob. 3 (Lists)	Prob. 4 (Stacks)	Prob. 5 (Trees)
Pre-Test	0.334 (p=0.001)	0.214 (p=0.017)	0.269 (p=0.009)
Total Time	0.186 (p=0.022)	0.125 (p=0.076)	0.129 (p=0.085)
Total Turns	0.129 (p=0.061)	0.134 (p=0.065)	
Draw Turns	0.116 (p=0.076)	0.122 (p=0.080)	
Code Turns		0.130 (p=0.071)	

Table 2: Problem Score Predictors (R^2)

lem solving, potentially there there is a relation between these participation correlations and initiative.

An analysis of initiative shows that there is a correlation of initiative and successful collaboration. In problem 3, learning gain positively correlates with the number of turns where a student has initiative ($R^2 = 0.156, p = 0.037$). And in problem 4, taking initiative through drawing has a positive impact on post-test score ($R^2 = 0.155, p = 0.047$).

3 Annotation

Since the preliminary analysis showed a correlation of initiative with learning gain, I chose to begin a thorough data analysis by annotating the dialogues with initiative shifts. Walker and Whittaker claim that initiative encompasses both dialogue control and task control (Walker and Whittaker, 1990), however, several others disagree. Jordan and Di Eugenio propose that control and initiative are two separate features in collaborative problem solving dialogues (Jordan and Di Eugenio, 1997). While control and initiative might be synonymous for the dialogues analyzed by Walker and Whittaker where a master-slave assumption holds, it is not the case in collaborative dialogues where no such assumption exists. Jordan and Di Eugenio argue that the notion of control should apply to the dialogue level, while initiative should pertain to the problem-solving goals. In a similar vein, Chu-Carroll and Brown also argue for a distinction between control and initiative, which they term task initiative and dialogue initiative (Chu-Carroll and Brown, 1998). Since there is no universally agreed upon definition for initiative, I have decided to annotate for both dialogue initiative and task initiative. For dialogue initiative annotation, I am using Walker and Whittaker's utterance based allocation of control rules (Walker and Whittaker, 1990), which are widely used to identify dialogue initiative. For task initiative, I have derived an annotation scheme based on other research in the area. According to Jordan and Di Eugenio, in problem solving (task) initiative the agent takes it upon himself to address domain goals by either (1)proposing a solution or (2)reformulating goals. In a similar vein, Guinn (Guinn, 1998) defines task initiative as belonging to the participant who dictates which decomposition of the goal will be used by both participants during problem-solving. A third definition is from Chu-Carroll and Brown. They suggest that task initiative tracks the lead in development of the agent's plan. Since the primary goal of the dialogues studied by Chu-Carroll and Brown is to develop a plan, this could be re-worded to state that task initiative tracks the lead in development of the agent's goal. Combining these definitions, task initiative can be defined as *any action by a participant to either achieve a goal directly, decompose a goal or reformulate a goal*. Since the goals of our problems are understanding and potentially correcting a program, actions in our domain that show task initiative include actions such as explaining what a section of code does or identifying a section of code that is incorrect.

Two coders, the author and an outside annotator, have coded 24 dialogues (1449 utterances) for both dialogue and task initiative. This is approximately 45% of the corpus. The resulting intercoder reliability, measured with the Kappa statistic, is 0.77 for dialogue initiative annotation and 0.68 for task initiative, both of which are high enough to support tentative conclusions. Using multiple linear regression analysis on these annotated dialogues, I found that, in a subset of the problems, there was a signicant correlation between post-test score (after removing the effects of pre-test scores) and the number of switches in dialogue initiative (R^2 =0.157, p=0.014). Also, in the same subset, there was a correlation between post-test score and the number of turns that a student had initiative (R^2 =0.077, p=0.065). This suggests that both taking the initiative and taking turns in leading problem solving results in learning.

Given my hypothesis that initiative can be used to identify co-construction, the next step is to annotate the dialogues using a subset of the DAMSL scheme (Core and Allen, 1997) to identify episodes of co-construction. Once annotated, I will use machine learning techniques to identify co-construction using initiative as a feature. Since this is a classification problem, algorithms such as Classification Based on Associations (Liu, 2007) will be used. Additionally, I will explore those algorithms that take into account the sequence of actions, such as hidden Markov models or neural networks.

4 Computational Model

The model will be implemented as an artificial agent, KSC-PaL, that interacts with a peer in collaborative problem solving using an interface similar to the one that was used in data collection (see Figure 1). This agent will be an extension of the TuTalk system, which is designed to support natural language dialogues for educational applications (Jordan et al., 2006). TuTalk contains a core set of dialogue system modules that can be replaced or enhanced as required by the application. The core modules are understanding and generation, a dialogue manager which is loosely characterized as a finite state machine with a stack and a student model. To implement the peer agent, I will replace TuTalk's student model and add a planner module.

Managing the information state of the dialogue (Larsson and Traum, 2000), which includes the beliefs and intentions of the participants, is important in the implementation of any dialogue agent. KSC-PaL will use a student model to assist in management of the information state. This student model tracks the current state of problem solving as well as estimates the student's knowledge of concepts involved in solving the problem by incorporating problem solution graphs (Conati et al., 2002). Solution graphs are Bayesian networks where each node represents either an action required to solve the problem or a concept required as part of problem solving. After analyzing our dialogues, I realized that the solutions to the problems in our domain are different from standard problem-solving tasks. Given that our tasks are program comprehension tasks and that the dialogues are peer led, there can be no assumption as to the order in which a student will analyze code statements. Therefore a graph comprised of connected subgraphs that each represent a section of the code more closely matches what I observed in our dialogues. So, we are using a modified version of solution graphs that has clusters of nodes representing facts that are relevant to the problem. Each cluster contains facts that are dependent on one another. For example, one cluster represents facts related to the push method for a stack. As the code is written, it would be impossible to comprehend the method without understanding the prefix notation for incrementing. A user's utterances and actions can then be matched to the nodes within the clusters. This provides the agent with information related to the student's knowledge as well as the current topic under discussion.

A planner module will be added to TuTalk to provide KSC-PaL with a more sophisticated method of selecting scripts. Unlike TuTalk's dialogue manager which uses a simple matching of utterances to concepts in order to determine the script to be followed, KSC-PaL's planner will incorporate the results of the data analysis above and will also include the status of the student's knowledge, as reflected in the student model, in making script selections. This planner will potentially be a probabilistic planner such as the one in (Lu, 2007).

5 Conclusion

In conclusion, we are developing a computational model of knowledge construction which incorporates initiative and the balance of initiative. This model will be embedded in an artificial agent that collaborates with students to solve data structure problems. As knowledge construction has been shown to promote learning, this research could have a profound impact on educational applications by changing the way in which they engage students in learning.

Acknowledgments

The graphical interface is based on a graphical interface developed by Davide Fossati for an intelligent tutoring system in the same domain.

References

Michelene T. H. Chi, Stephanie A. Siler, Jeong Heisawn, Takashi Yamauchi, and Robert G. Hausmann. 2001. Learning from human tutoring. *Cognitive Science*, 25(4):471–533.

Jennifer Chu-Carroll and Michael K. Brown. 1998. An evidential model for tracking initiative in collaborative dialogue interactions. *User Modeling and User-Adapted Interaction*, 8(3–4):215–253, September.

Jennifer Chu-Carroll and Sandra Carberry. 1998. Collaborative response generation in planning dialogues. *Computational Linguistics*, 24(3):355–400.

Cristina Conati, Abigail Gertner, and Kurt Vanlehn. 2002. Using bayesian networks to manage uncertainty in student modeling. *User Modeling and User-Adapted Interaction*, 12(4):371–417.

María de los Angeles Constantino-González and Daniel D. Suthers. 2000. A coached collaborative learning environment for entity-relationship modeling. *Intelligent Tutoring Systems*, pages 324–333.

Mark G. Core and James F. Allen. 1997. Coding dialogues with the DAMSL annotation scheme. In David Traum, editor, *Working Notes: AAAI Fall Symposium on Communicative Action in Humans and Machines*, pages 28–35, Menlo Park, California. American Association for Artificial Intelligence.

Martha W. Evens, Ru-Charn Chang, Yoon Hee Lee, Leem Seop Shim, Chong Woo Woo, Yuemei Zhang, Joel A. Michael, and Allen A. Rovick. 1997. Circsim-tutor: an intelligent tutoring system using natural language dialogue. In *Proceedings of the fifth conference on Applied natural language processing*, pages 13–14, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Arthur C. Graesser, Shulan Lu, George Tanner Jackson, Heather Hite Mitchell, Mathew Ventura, Andrew Olney, and Max M. Louwerse. 2004. Autotutor: A tutor with dialogue in natural language. *Behavior Research Methods, Instruments, & Computers*, 36:180–192(13), May.

Curry I. Guinn. 1998. An analysis of initiative selection in collaborative task-oriented discourse. *User Modeling and User-Adapted Interaction*, 8(3-4):255–314.

Robert G.M. Hausmann, Michelee T.H. Chi, and Marguerite Roy. 2004. Learning from collaborative problem solving: An analysis of three hypothesized mechanisms. In K.D Forbus, D. Gentner, and T. Regier, editors, *26th Annual Converence of the Cognitive Science Society*, pages 547–552, Mahwah, NJ.

Pamela W. Jordan and Barbara Di Eugenio. 1997. Control and initiative in collaborative problem solving dialogues. In *Working Notes of the AAAI Spring Symposium on Computational Models for Mixed Initiative*, pages 81–84, Menlo Park, CA.

Pamela W. Jordan, Michael Ringenberg, and Brian Hall. 2006. Rapidly developing dialogue systems that support learning studies. In *Proceedings of ITS06 Workshop on Teaching with Robots, Agents, and NLP*, pages 1–8.

Staffan Larsson and David R. Traum. 2000. Information state and dialogue management in the trindi dialogue move engine toolkit. *Nat. Lang. Eng.*, 6(3-4):323–340.

Bing Liu. 2007. *Web data mining: exploring hyperlinks, contents, and usage data.* Springer.

Karen E. Lochbaum and Candice L. Sidner. 1990. Models of plans to support communication: An initial report. In Thomas Dietterich and William Swartout, editors, *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 485–490, Menlo Park, California. AAAI Press.

Xin Lu. 2007. *Expert Tutoring and Natural Language Feedback in Intelligent Tutoring Systems*. Ph.D. thesis, University of Illinois at Chicago.

Amy Soller. 2004. Computational modeling and analysis of knowledge sharing in collaborative distance learning. *User Modeling and User-Adapted Interaction*, Volume 14(4):351–381, January.

Tan Bee Tin. 2003. Does talking with peers help learning? the role of expertise and talk in convergent group discussion tasks. *Journal of English for Academic Purposes*, 2(1):53–66.

Kurt VanLehn, Pamela W. Jordan, Carolyn Penstein Rosé, Dumisizwe Bhembe, Michael Böttner, Andy Gaydos, Maxim Makatchev, Umarani Pappuswamy, Michael A. Ringenberg, Antonio Roque, Stephanie Siler, and Ramesh Srivastava. 2002. The architecture of why2-atlas: A coach for qualitative physics essay writing. In *ITS '02: Proceedings of the 6th International Conference on Intelligent Tutoring Systems*, pages 158–167, London, UK. Springer-Verlag.

Aurora Vizcaíno. 2005. A simulated student can improve collaborative learning. *International Journal of Artificial Intelligence in Education*, 15(1):3–40.

Marilyn Walker and Steve Whittaker. 1990. Mixed initiative in dialogue: an investigation into discourse segmentation. In *Proceedings of the 28th annual meeting on Association for Computational Linguistics*, pages 70–78, Morristown, NJ, USA. Association for Computational Linguistics.

Kim Onn Yap. 1979. Pretest-posttest correlation and regression models. Presented at the Annual Meeting of the American Educational Research Association (63rd, San Francisco, California), April 8–12.