Modeling Knowledge Co-Construction for Peer Learning Interactions *

Cynthia Kersey[1], Barbara Di Eugenio[1], Pamela Jordan[2], and Sandra Katz[2]

[1] University of Illinois at Chicago, Computer Science Department
851 S. Morgan (M/C 152), Room 1120 SEO
Chicago, IL 60607-7053, United States
[2] University of Pittsburgh, Learning Research and Development Center
3939 O'Hara St., Rm. 744 LRDC
Pittsburgh, PA 15260, United States

Abstract. We are analyzing peer collaborations in order to build a computational model of knowledge co-construction that could be useful in creating an effective artificial peer learning agent. We hypothesize that the start of a co-construction episode can be predicted based on initiative during interactions and that shifts of initiative during interactions indicate that co-construction is taking place. Since knowledge construction during collaboration is thought to be beneficial to individuals and dyads, in this paper we show preliminary results indicating that initiative and shifts of initiative are correlated with the learning gains and task performance of individuals and of dyads.

1 Introduction

Peer tutoring and learning have been shown to strongly promote learning [1–4]. Students working together have more frequent generation of new ideas and a higher level of reasoning [5]. There are various theories as to why collaboration in peer learning is effective, but one that is commonly referenced is co-construction [6]. 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. Essentially, students who are active in the learning process are more successful. Examples of co-constrution are a peer adding to or extending a partner's contribution or critically evaluating a partner's input.

Peer learning agents attempt to embody notions of co-construction to various degrees. Some take on the role of a companion instead of a more authoritative tutor [7]. Some peer agents behave as coaches and monitor the student interaction or provide help in problem solving [8]. Other peer learning agents act as tutees which are taught or coached by a student [9–11]. Additional peer learning agents encourage reciprocal tutoring and can play either the role of tutor or tutee [12, 13]. However, while the agent can take on different roles in different interactions, the role is fixed for the duration of the interaction.

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

Our goal is an innovative peer agent that can switch roles during the interaction in order to encourage knowledge co-construction. Clearly, the agent must be able to accurately identify when a student is engaged in co-construction. Additionally, since a peer interaction can sometimes go awry [12, 14–17], in those cases we would like the agent to put the interaction back on track without stifling the possibility of co-constructive behavior returning later on in the interaction. Hence, we also need to identify behaviors which tend to stifle co-construction, so the peer agent can avoid them

Instances of co-construction appear to closely resemble two types of initiative that the dialogue research community distinguishes between; dialogue initiative and task initiative[18]. Dialogue initiative tracks who is leading the conversation and determining the current conversational focus while task initiative tracks the leader in the development of a plan to achieve a problem solving goal. Thus we hypothesize that dialogue and task initiative will aid in predicting the beginning of knowledge construction episodes and that shifts in who is taking initiative will indicate that co-construction is taking place. Shifts in initiative show that participants in the interaction are having the chance to both lead and criticize or elaborate on an extension to the problem solving plan.

To explore these hypotheses, we analyzed a corpus of student collaborations in which dyads are working on tasks that are meant to increase their understanding of computer science data structures. 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.

The results of this analysis will be used to develop a computational model that will be embedded in an artificial peer learning agent. We have begun work on the agent, with the development of an interface and a student model that will track the current state of problem solving as well as estimate the student's knowledge of concepts involved in solving the problem.

In this paper we present our data collection and an analysis of initiative's impact on successful collaboration and learning. We conclude with a discussion of future work.

2 Data Collection

To test our hypothesis about the relationship between initiative and knowledge co-construction during peer collaborations, we analyzed a corpus of interactions between 15 student dyads working together via a computer mediated interface to solve five data structure problems. The data structures that we are focusing on are (1) linked lists, a set of data nodes connected by pointers (see example in Fig. 1 drawing), (2) stacks, array-based last-in first-out structures and (3) binary search trees, a data structure where each node connects to at most two other nodes and whose ordering property makes it very efficient for retrieving data. The participants in these interactions were undergraduate computer science students who were taking or had taken at least one course in data structures. The problems presented to the students required them to analyze and potentially correct code segments involving data structures. The computer-mediated interface was designed after first observing face-to-face interactions of students solving these problems. Because the students used other communicative channels in addition to dialogue, such as drawing diagrams, the interface consists of four distinct areas (see Figure 1):

1. Problem display: Presents the problem that the students are to solve.
2. Code display: Displays the code from the problem statement. Additionally, 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. This area functions similarly to an instant messaging application.
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 be placed in the drawing area to build lists, stacks or trees depending on the type of problem being solved. Such diagraming is widely used in data structure courses, so students are familiar with the concept of building data structures using such objects.

The changes made in the shared workspace (drawing and code areas) are logged and propagated to the partner's window. This gives the users the ability to communicate not only verbally but also via graphical actions.

Before and after their collaborative problem solving interactions, each student was individually given a test consisting of ten data structure problems, similar to the problems presented to them during collaborative problem solving. On completion of the pre-test, the students were given a short tutorial on the interface and then seated at computers in separate rooms. Each pair was given five problems to solve using the computer-mediated interface. Problems 1, 2 and 3 involved linked list data structures, while problem 4 was a stack problem and problem 5 related to binary search trees. The initial exercise let the users become acquainted with the interface. They 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 the pre-test and post-test could not exceed three hours. When solving the problems, the participants interacted via the computer interface through typed natural language utterances and through actions in the shared workspace as shown in the transcript sample (see Figure 2). In this transcript the dyad is working on problem 1. After discussing a misconception of C's, in 14:03:40 C begins to illustrate the effect of the first line of code in the code sample given to the pair. First a pointer variable is added to the drawing and then in 14:03:47 that variable is made to point at a node in the drawing.

Given the time constraint, not all pairs completed all five problems. Thus we have a corpus of 69 problem solving dialogues.

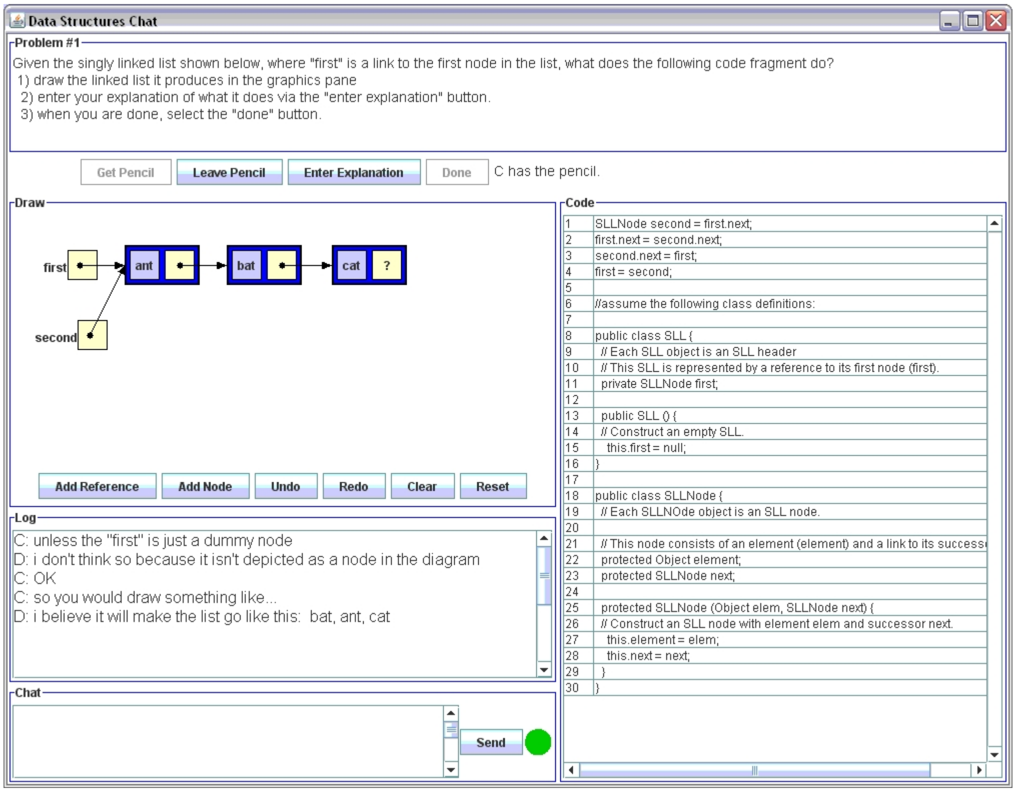

Fig. 1. The data collection interface

3 Data Analysis

A paired t-test of pre- and post-test scores showed that *students did learn* during collaborative problem solving (t(30)=2.83; p=0.007). Additionally the interactions produced an average normalized learning gain of 17.5.

We then performed an initial analysis on three of the exercises (problems 3, 4 and 5) to identify features that positively impacted learning and problem solving. Since both chat and shared workspace actions were logged for each user, we were able to automatically extract the following features for each exercise (in parentheses we list the labels we will use in reporting results in tables):

- Total number of turns (total turns)
- Drawing turns (drawing turns)
- Actual drawing turns that exclude those that only rearranged the drawing (actual drawing turns)
- Code turns (code turns)

```
14:01:56 C:  unless the "first" is just a dummy node
14:02:20 D:  i don't think so because it isn't depicted
             as a node in the diagram
14:02:28 C:  OK
14:03:13 C:  so you would draw something like...
14:03:24 D:  i believe it will make the list go like this:
             bat, ant, cat
14:03:40 C:  draw: add pointer second (n100)
14:03:44 C:  draw: move n100
14:03:46 C:  draw: link n100 to null
14:03:47 C:  draw: link n100 to n002
```

Fig. 2. An excerpt from one of the interactions

- Time spent on graphical actions (time on graphical actions)
- Total number of words (words)
- Words per turn (words per turn)
- Total problem solving time (total time)

Additionally, all pairs received a score on the solutions that they submitted (problem score). For each problem completed, a pair could earn a maximum of five points based on the correctness and completeness of their solution. Since each pair was presented with 5 problems, they could earn a total of 25 points.

Linear regression analysis revealed significant correlations and trends toward correlations between some of these features and learning and successful problem solving (see Table 1 and Table 2). In problems 3 there is a trend toward correlation of drawing with both post-test score and problem score and in problem 4 there is a correlation with problem score. This suggests that use of the graphical workspace was beneficial to the students. The remaining correlations and trends to correlation also suggest that participation in general is an important factor in collaborative learning and problem solving.

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

Predictor	**Problem 3 (Lists)**	**Problem 4 (Stacks)**	**Problem 5 (Trees)**
Pre-Test	0.336 (p=0.001)	0.657 (p=0.000)	0.663 (p=0.000)
Words	0.189 (p=0.021)		
Words per Turn	0.141 (p=0.049)		
Time on graphical actions	0.154 (p=0.039)		
Problem Score	0.315 (p=0.002)		
Total Turns	0.108 (p=0.088)		
Actual Drawing Turn	0.105 (p=0.092)		
Code Turns			0.136 (p=0.076)

Table 2. Problem Score Predictors (R^2)

Predictor	Problem 3 (Lists)	Problem 4 (Stacks)	Problem 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)	
Drawing Turns	0.116 (p=0.076)	0.122 (p=0.080)	
Actual Drawing Turns		0.124 (p=0.078)	
Draws			0.159 (p=0.054)
Code Turns		0.130 (p=0.071)	

Our analysis of these basic, easily extractable features shows that there is some correlation between participation and our measures of successful collaboration (post-test score and problem score) and justifies doing a deeper analysis of the collaborations. We chose to annotate for initiative because we believe that it can aid in the identification of knowledge co-construction episodes. Intuitively, initiative would switch between peers when they are working together to construct a solution and solve the problem. Since there are various types and definitions for initiative [18–23], we chose to annotate for two different types.

First, the dialogues were annotated for dialogue initiative, which tracks control over the conversation, using Walker and Whittaker's utterance based rules for attributing control [23]. In this scheme, each turn in the dialogue must first be tagged as either: (1) an assertion, (2) a command, (3) a question or (4) a prompt (a turn not expressing propositional content). Control is then attributed using rules based on the turn type. Since these rules only include dialogue actions, graphical actions were excluded from this annotation.

Annotation for task initiative included graphical actions as well as chat actions. We define task initiative as taking the lead in problem solving activities. Actions in our domain that show task initiative include:

- Suggesting a section of code to verify.
- Explaining what a section of code does.
- Identifying that a section of code as correct or incorrect.
- Suggesting a correction to a section of code
- Making a correction to a section of code prior to discussion with the other participant.

Two coders, one of the authors and an outside annotator, have coded 24 dialogues (1449 utterances) for both types of initiative. This is approximately 45% of the corpus. The resulting intercoder reliability, measured with the Kappa statistic[24] (shown in table 3) is high enough to support tentative conclusions.

To determine if initiative has a correlation with learning, multiple linear regressions were run with post-test score as the predicted variable. Predictor variables include the students' prior knowledge (pre-test score), the number of turns that a student had initiative and the number of initiative switches between the participants. Separate regressions were run for each of the problem

Table 3. Kappa Values for Initiative Annotation

	Dialogue Initiative	Task Initiative
Kappa	0.77	0.68

types: list (problems 2 and 3), stack(problem 4) and trees (problem 5) as well as combinations of the different problem types. Problem 1 was excluded from the analysis since its purpose was to let the participants become familiar with the interface.

In the list problems, we found that there was a significant 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, as shown in table 4, there was also a correlation between post-test score and the number of turns that a student had initiative. This suggests that learning increases when students often take the initiative and also when they take turns leading problem solving. Additionally, multiple regressions were run using problem solving success (problem score) as the predicted variable. Predictor variables were the same as for the other regression analyses, except since problem score is for a dyad, prior knowledge is measured as the higher of the pairs' pre-test scores. For list and stack problems combined, after regressing out the maximum pre-test score, the number of task initiative switches correlates with problem score(R^2=0.257, p=0.052).

Table 4. Multiple Regression Results

Problems	Predicted Variable	Predictor Variable	R^2	β	p
2-3	Post-test Score	Pre-test Score	0.324	0.559	0.001
		Dialogue Initiative Switches	0.157	0.382	0.014
2-3	Post-test Score	Pre-Test Score	0.345	0.596	0.001
		Dialogue Initiative Turns	0.077	0.294	0.065
2-4	Problem Score	Maximum Pre-test Score	0.407	0.563	0.024
		Task Initiative Switches	0.257	0.410	0.052

Several factors potentially play into the fact that not all problem types showed a correlation of initiative features with measures of successful collaboration. First, the lack of correlations in the tree problem is possibly caused by the wide variation in experience levels of the students. Of the pairs that solved the tree problem, only 33% had both members receiving an acceptable score (more than 60% of the possible points) on the tree related problems in the pre-test. This contrasts with 58% for the list problems and 68% for the stack problems. And secondly, since the students had a better understanding of stacks prior to problem solving, there was less discussion in solving the stack problems. Additionally, our experience in teaching data structures in the classroom is that students struggle more with the concepts related to linked lists than with those involved in understanding stacks. So, a better overall understanding of stacks is a possible cause of the lack of correlation of dialogue initiative with post-test score in the stack problems.

Based on these results, our next steps will be to test whether knowledge co-construction correlates with initiative and initiative shifts and to build a model to identify initiative automatically.

4 Current and Future Work

One issue we have not settled on yet is whether initiative, as we have coded so far, is sufficient to identify co-construction. Intuitively it is not, and one would want more sophisticated and telling predictors. However, we must take into account the difficulty of recognizing such predictors while keeping in mind our ultimate goal of having a software agent recognize them.

For example, we have started annotating for relations that could potentially identify knowledge construction. In an initial attempt, two coders have annotated a subset of the dialogues for the following relations:

- criticize: a student critically evaluates her peer's input
- elaborate: a student adds additional information to the topic under discussion
- justify: a student adds support to a statement made by a peer
- summarize: a student recaps the discussion related to a segment of code

The resulting intercoder reliability for these relations is not good (see Table 5). These values reflect the difficulties that humans have in identifying such relations. While we are trying to improve this annotation, we are also asking whether it is worth pursuing or whether we can use initiative annotation alone since we have found initiative to be much easier to identify. As shown in Tables 3 and 5, with the exception of the justification relation, dialogue initiative and task initiative have greater inter-coder reliability. A correlation with initiative would make co-construction episodes easier to identify.

Table 5. Kappa Values for Co-Construction Relations

	Criticize	Elaborate	Justify	Summarize
Kappa	0.5	0.13	0.75	0.20

Given our hypothesis that initiative can at a minimum aid in the identification of co-construction, initial work has begun on creating a model to automatically identify initiative based on easily attainable features of the interactions. We are currently exploring various machine learning algorithms. Since this is a classification problem, we will investigate classifiers, such as Decision Trees and Classification Based on Associations[25]. However, those algorithms might not capture the sequential nature of dialogue, so algorithms such as as hidden

Markov models or neural networks that take into account the sequence of actions might be a better fit to our data.

Once our model is developed, it 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 [26]. Such an agent would be used to augment in-class instruction by providing students with the opportunity to solve additional problems with the help of a "peer".

5 Conclusion

In this paper we've presented our intial steps toward creating a model that can successfully recognize co-construction and the lack thereof. Because we expect knowledge construction to be beneficial to learning and problem solving, we tested whether initiative, which is defined similarly to knowledge co-construction but simpler to recognize, is correlated with learning and task performance and found that it is. Our next step is to annotate directly for co-construction episodes and check that they are correlated with initiative. A model that can recognize initiative could be used to intervene when a student collaborator is not engaging in co-construction. Such a model will be embedded in an agent that would interact with users using both typed natural language dialogue and actions in a graphical workspace.

References

1. Cohen, P., Kulik, J., Kulik, C.: Education outcomes of tutoring: A meta-analysis of findings. American Education Research Journal **19**(2) (1982) 237–248
2. Brown, A.L., Palinscar, A.S. In: Guided, cooperative learning and individual knowledge acquisition. (1989) 307–226
3. Birtz, M.W., Dixon, J., McLaughlin, T.F.: The effects of peer tutoring on mathematics performance: A recent review. B. C. Journal of Special Education **13**(1) (1989) 17–33
4. Rekrut, M.D.: Teaching to learn: Cross-age tutoring to enhance strategy instruction. American Education Research Association (1992)
5. Johnson, D.W., Johnson, R.T.: Learning Together and Alone. (1999)
6. Hausmann, R.G., Chi, M.T., Roy, M.: Learning from collaborative problem solving: An analysis of three hypothesized mechanisms. (2004)
7. Chou, C.Y., Chan, T.W., Lin, C.J.: Redefining the learning companion: the past, present, and future of educational agents. Comput. Educ. **40**(3) (2003) 255–269
8. Vizcaíno, A.: A simulated student can improve collaborative learning. International Journal of Artificial Intelligence in Education **15**(1) (2005) 3–40
9. Baylor, A.L.: Cognitive requirements for agent-based learning environments. Second IEEE International Conference on Advanced Learning Technologies (ICALT'01), **00** (2001) 0462

10. Biswas, G., Katzlberger, T., Bransford, J., Schwartz, D.: Extending intelligent learning environments with teachable agents to enhance learning. Artificial Intelligence in Education (2001) pp. 389–397
11. Urestí, J.A.R.: Should i teach my computer peer? some issues in teaching a learning companion. (2000) 103–112
12. Chou, C.Y., Lin, C.J., Chan, T.W.: An approach to developing computational supports for reciprocal tutoring. Knowledge-Baed Systems **15**(7) (September 2002) 407–412
13. Reif, F., Scott, L.A.: Teaching scientific thinking skills: Students and computers coaching each other. American Journal of Physics **67**(9) (1999) 819–831
14. Constantino-González, M.d., Suthers, D.D.: Coaching collaboration by comparing solutions and tracking participating. In: Proceedings of the 1st European Convergence in Computer Supported Collavorative Learning EURO-CSCL, Maastricht, the Netherlands (2001) 704–705
15. Ramirez Urestí, J.A.: Should i teach my computer peer? some issues in teaching a learning companion. In: Fifth International Conference on Intelligent TUtoring SYstems (ITS'2000), Montreal, Canada (2000) 103–112
16. Soller, A.: Computational modeling and analysis of knowledge sharing in collaborative distance learning. User Modeling and User-Adapted Interaction **Volume 14**(4) (January 2004) 351–381
17. VanLehn, K., Ohlsson, S., Nason, R.: Applications of simulated students: An exploration. Journal of Artificial Intelligence in Education **5**(2) (1994) 135–175
18. Chu-Carroll, J., Brown, M.K.: An evidential model for tracking initiative in collaborative dialogue interactions. User Modeling and User-Adapted Interaction **8**(3–4) (September 1998) 215–253
19. Cohen, R., Allaby, C., Cumbaa, C., Fitzgerald, M., Ho, K., Hui, B., Latulipe, C., Lu, F., Moussa, N., Pooley, D., Qian, A., Siddiqi, S.: What is initiative? User Modeling and User-Adapted Interaction **8**(3-4) (1998) 171–214
20. Heeman, P.A., Yang, F., .Strayer, S.E.: Control in task-oriented dialogues. In: EUROSPEECH-2003. (2003) 209–212
21. Jordan, P., Siler, S.: Student initiative and questioning strategies in computer-mediated human tutoring dialogues. In: Proceedings of ITS 2002 Workshop on Empirical Methods for Tutorial Dialogue Systems. (2002)
22. Jordan, P.W., Di Eugenio, B.: Control and initiative in collaborative problem solving dialogues. In: Working Notes of the AAAI Spring Symposium on Computational Models for Mixed Initiative, Menlo Park, CA (1997) 81–84
23. Walker, M., Whittaker, S.: Mixed initiative in dialogue: an investigation into discourse segmentation. (1990) 70–78
24. Carletta, J.: Assessing agreement on classification tasks: the kappa statistic. Comput. Linguist. **22**(2) (1996) 249–254
25. Liu, B.: Web data mining: exploring hyperlinks, contents, and usage data. (2007)
26. Jordan, P.W., Ringenberg, M., Hall, B.: Rapidly developing dialogue systems that support learning studies. In: Proceedings of ITS06 Workshop on Teaching with Robots, Agents, and NLP. (2006) 1–8