KSC-PaL: A Peer Learning Agent that Encourages Students to take the Initiative*

Cynthia Kersey and Barbara Di Eugenio
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607 USA
ckerse2@uic.edu
bdieugen@cs.uic.edu

Pamela Jordan and Sandra Katz
Learning Research and Development Center
University of Pittsburgh
Pittsburgh, PA 15260 USA
pjordan+@pitt.edu
katz+@pitt.edu

Abstract

We present an innovative application of discourse processing concepts to educational technology. In our corpus analysis of peer learning dialogues, we found that initiative and initiative shifts are indicative of learning, and of learning-conducive episodes. We are incorporating this finding in KSC-PaL, the peer learning agent we have been developing. KSC-PaL will promote learning by encouraging shifts in task initiative.

1 Introduction

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), however, the study of peer learning from a computational perspective is still in the early stages. This is an important area of study because peer learning has been shown to be an effective mode of learning, potentially for all of the participants (Cohen et al., 1982; Brown and Palincsar, 1989; Birtz et al., 1989; Rekrut, 1992). 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.

Previous research has suggested several mechanisms that explain why peer learning is effective for all participants. Among them are: self-directed explaining(Chi et al., 1994), other-directed explaining (Ploetzner et al., 1999; Roscoe and Chi, 2007) and Knowledge Co-construction – KCC for short (Hausmann et al., 2004). KCC episodes are defined as portions of the dialogue in which students are jointly constructing a shared meaning of a concept required for problem solving. This last mechanism is the most interesting from a peer learning perspective because it is a truly collaborative construct and also because it is consistent with the widely accepted constructivist view of learning.

Since KCC is a high-level concept that is not easily recognized by an artificial agent we collected peer learning interactions from students and studied them to identify features that might be useful in identifying KCC. We found that linguistically based initiative shifts seem to capture the notion of collaborative construction. A more thorough analysis found a strong relationship between KCC and initiative shifts and moderate correlations between initiative shifts and learning.

The results of this analysis are being incorporated into KSC-PaL, an artificial agent that can collaborate with a human student via natural-language dialogue and actions within a graphical workspace. KSC-PaL has been developed in the last two years. Dialogue-wise, its core is TuTalk (Jordan et al., 2007), a dialogue management system that supports natural lan-

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

guage dialogue in educational applications. As we will describe, we have already developed its user interface and its student model and have extended TuTalk's planner to provide KSC-PaL with the ability to induce initiative shifts. For the version of KSCPal we will present in this paper, we wanted to focus on the question of whether this style of interaction helps learning; and we were concerned that its limitations in disambiguating the student's input could impact this interaction. Hence, this round of experiments employs a human "helper" that is given a list of concepts the input may match, and chooses the most appropriate one.

The work presented in this paper is part of a larger research program: we analyze different paradigms – tutoring dialogues and peer-learning dialogues– in the same basic domain, devise computational models for both, and implement them in two separate SW systems, an ITS and the peer-learning system we present here. For our work on the tutoring dialogue corpus and the ITS please see (Fossati et al., accepted for publication 2009).

Our domain in both cases is problem solving in basic data structure and algorithms, which is part of foundations of Computer Science. While in recent years, interest in CS in the US has dropped dramatically, CS is of enormous strategic interest, and is projected to foster vast job growth in the next few years (AA. VV., 2006). We believe that by supporting CS education in its core we can have the largest impact on reversing the trend of students' disinterest. Our belief is grounded in the observation that the rate of attrition is highest at the earliest phases of undergraduate CS curricula. This is due in part to students' difficulty with mastering basic concepts (Katz et al., 2003), which require a deep understanding of static structures and the dynamic procedures used to manipulate them (AA. VV., 2001). These concepts also require the ability to move seamlessly among multiple representations, such as text, pictures, pseudo-code, and real code in a specific programming language.

Surprisingly, few educational SW systems address CS topics, e.g. teaching a specific programming language like LISP (Corbett and Anderson, 1990) or database concepts (Mitrović et al., 2004). Additionally, basically they are all ITSs, where the relationship between the system and the student is one of "subordination". Only two or three of these ITSs address foundations, including: Autotutor (Graesser et al., 2004) addresses basic literacy, but not data structures or algorithms; ADIS (Warendorf and Tan, 1997) tutors on basic data structures, but its emphasis is on visualization, and it appears to have been more of a proof of concept than a working system; ProPL (Lane and VanLehn, 2003) helps novices design their programs, by stressing problem solving and design skills.

In this paper, we will first discuss the collection and analysis of peer learning interactions. Then, we discuss the design of our peer agent, and how it is guided by the results of our analysis. We conclude by briefly describing the user experiments we are about to undertake, and whose preliminary results will be available at the time of the workshop.

2 Data collection

We have collected peer learning interactions from 15 pairs of students solving problems in the domain of computer science data structures. Students were recruited from introductory courses on data structures and algorithms. Each problem involved one of three types of data structures: linked-lists, stacks and binary search trees. Each problem was either a debugging problem where the students were asked to work together to identify errors in the code or an explanation problems in which the students jointly created an explanation of a segment of code.

The students interacted using a computer mediated interface[1] where they could communicate via text-based chat, drawing and making changes to code (see Figure 1). The graphical workspace (drawing and coding areas) was shared such that changes made by one student were propagated to his/her partner's workspace. Access to this graphical workspace was controlled so that only one student was allowed to draw or make changes to code at any point in time.

Each pair was presented with a total of 5 problems, although not all pairs completed all problems due to time limitations. The interactions for each pair were subdivided into separate dialogues

[1]Using text to communicate versus face-to-face interactions should be comfortable for most students given the prevalence of communication methods such as text messaging and instant messengers.

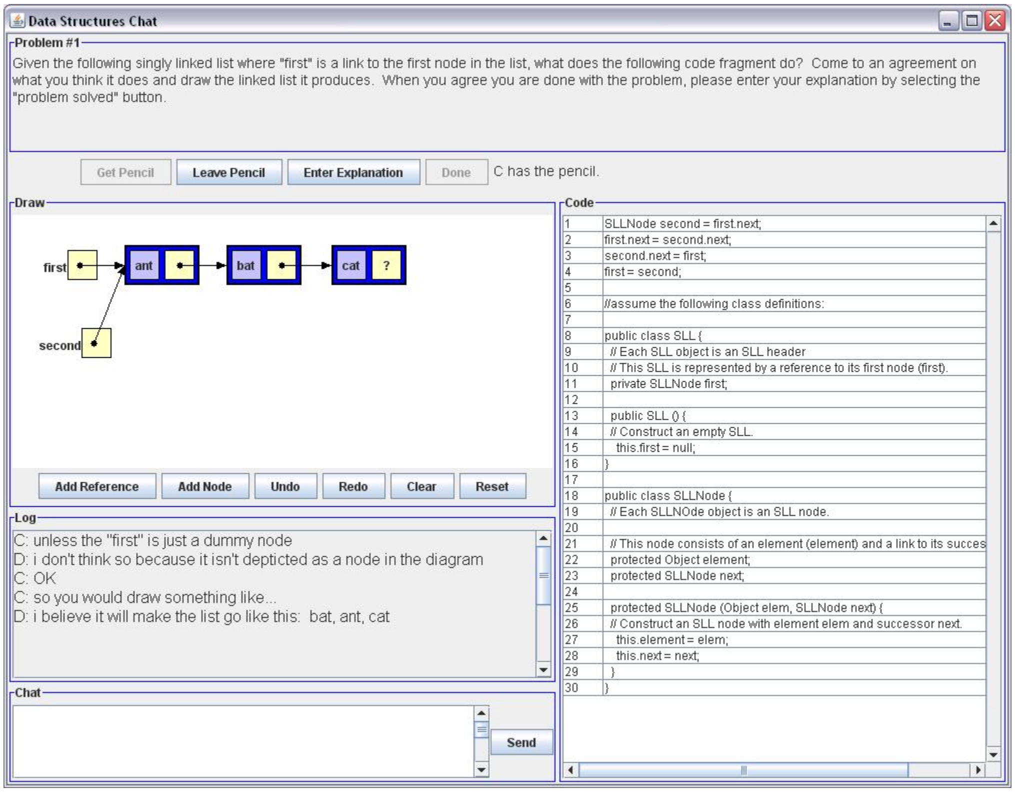

Figure 1: The data collection / KSC-PaL interface

for each problem. Thus, we collected a corpus consisting of a total of 73 dialogues.

In addition to collecting problem solving data, we also presented each student with a pre-test prior to problem solving and an identical post-test at the conclusion of problem solving in order to measure learning gains. 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). The interactions produced an average normalized learning gain of 17.5 (possible total points are 50).

3 Analysis of Peer Learning Interactions

Next, we undertook an extensive analysis of the corpus of peer learning interactions in order to determine the behaviors with which to endow KSC-PaL.

3.1 Initiative: Annotation

Given the definition of KCC, it appeared to us that the concept of initiative from discourse and dialogue processing should play a role: intuitively, if the students are jointly contructing a concept, the initiative cannot reside only with one, otherwise the partner would just be passive. Hence, we annotated the dialogues for both KCC and initiative.

The KCC annotation involved coding the dialogues for KCC episodes. These are defined as a series of utterances and graphical actions in which students are jointly constructing a shared meaning of a concept required for problem solving (Hausmann et al., 2004). Using this definition, an outside annotator and one of the authors coded 30 dialogues (approximately 46% of the corpus) for KCC episodes. This entailed marking the beginning utterance and the end utterance of such episodes, under the assumption that all intervening utterances do belong to the same KCC episode (otherwise the coder would mark an earlier end for the episode). The resulting intercoder reliability, measured with the Kappa statistic(Carletta, 1996), is considered excellent ($\kappa = 0.80$).

Our annotation of initiative was two fold. Since there is disagreement in the computational linguistics community as to the precise definition of initiative(Chu-Carroll and Carberry, 1998; Jordan and Di Eugenio, 1997), we annotated the dialogues for both dialogue initiative, which tracks who is leading the conversation and determining the current conversational focus, and task initiative, which tracks the lead in problem solving.

For dialogue initiative annotation, we used the well-known utterance-based rules for allocation of control from (Walker and Whittaker, 1990). In this scheme, each utterance is tagged with one of four dialogue acts (assertion, command, question or prompt) and control is then allocated based on a set of rules. The dialogue act annotation 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. To verify that the automatic annotation was good, we manually annotated a sizable portion of the dialogues with those four dialogue acts. We then compared the automatic annotation against the human gold standard, and we found an excellent accuracy: it ranged from 86% for assertions and questions, to 97% for prompts, to 100% for commands.

Once the dialogue acts had been automatically annotated, two coders, one of the authors and an outside annotator, coded 24 dialogues (1449 utterances, approximately 45% of the corpus) for dialogue initiative, by using the four control rules from (Walker and Whittaker, 1990):

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.

The resulting intercoder reliability on dialogue initiative was 0.77, a quite acceptable level of agreement. We then experimented with automatically annotating dialogue initiative according to those control rules. Since the accuracy against the gold standard was 82%, the remaining 55% of the corpus was also automatically annotated for dialogue initiative, using those four control rules.

As concerns task initiative, we define it as *any action by a participant to either achieve a goal directly, decompose a goal or reformulate a goal* (Guinn, 1998; Chu-Carroll and Brown, 1998). Actions in our domain that show task initiative include:

- 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.

The same two coders annotated for task initiative the same portion of the corpus already annotated for dialogue initiative. The resulting intercoder reliability for task initiative is 0.68, which is high enough to support tentative conclusions. The outside coder then manually coded the remaining 55% of the corpus for task initiative.

3.2 KCC, initiative and learning

In analyzing the annotated dialogues, we used multiple linear regression to identify correlations of the annotated features and post-test score. We used pre-test score as a covariate because of its significant positive correlations with post-test score. Due to variations in student ability in the different problem types, our analysis focused only on a portion of the collected interactions. In the tree problem there was a wide variation in experience level of the students which would inhibit KCC. In the stack problem, the students had a better understanding of stacks prior to problem solving and spent less time in discussion and problem solving. Thus, our analysis focused only on the linked-list problems.

We started by analyzing the relationship between KCC and learning. As a measurement of KCC we used *KCC actions* which is the number of utterances and graphical actions that occur during KCC episodes. This analysis showed that KCC does have a positive correlation with learning in our corpus. In Table 1, the first row shows the benefit for the dyad overall by correlating the mean post-test score with the mean pre-test score and the dyad's KCC actions. The second row shows the benefit for individuals by correlating individual post-test scores with individual pre-test scores and the dyad's KCC actions. The difference in the strength of these correlations suggests that members of the dyads are not benefitting equally from KCC. If the subjects are divided into two groups, those with a pre-test score below the mean score (*n*=14) and those with a pre-test score above the mean score (*n*=16) , it can be seen that those with a low pre-test score benefit more from the KCC episodes than do those with a high pre-test score (rows 3 and 4 in Table 1).

KCC actions predict	β	R^2	p
Mean post-test score	0.43	0.14	0.02
Individual post-test score	0.33	0.08	0.03
Individual post-test score (low pre-test subjects)	0.61	0.37	0.03
Individual post-test score (high pre-test subjects)	0.33	0.09	ns

Table 1: KCC Actions as Predictor of Post-test Score

Next, we explored the relationship between learning and the number of times initiative shifted between the students. Intuitively, we assumed that frequent shifts of initiative would reflect students working together to solve the problem. We found there was a significant correlation between post-test score (after removing the effects of pre-test scores) and the number of shifts in dialogue initiative and the number of shifts in task initiative (see Table 2). This analysis excluded two dyads whose problem solving collaboration had gone awry.

Predictor of Post-test	β	R^2	p
Dialogue initiative shifts	0.45	0.20	0.00
Task initiative shifts	0.42	0.20	0.01

Table 2: Initiative Predictors of Post-test Score

We then computed a second measure of KCC that is meant to reflect the density of the KCC episodes. *KCC initiative shifts* is the number of task initiative shifts that occur during KCC episodes. Many task initiative shifts reflect more active KCC.

Table 3 uses KCC initiative shifts as the measure of co-construction. It shows similar results to table 1, where KCC actions was used. Note that when the outlier dyads were removed the correlation with learning is much stronger for the low pre-test score subjects when KCC initiative shifts are used as the measure of KCC ($R^2 = 0.45$, $p = 0.02$) than when KCC actions are used.

KCC initiative shifts predict	β	R^2	p
Mean post-test score	0.46	0.15	0.01
Individual post-test score	0.35	0.09	0.02
Individual post-test score (low pre-test subjects)	0.67	0.45	0.02
Individual post-test score (high pre-test subjects)	0.10	0.01	ns

Table 3: KCC Initiative Shifts Predictors of Post-test Score

Lastly we investigated the hypothesis that KCC episodes involve frequent shifts in initiative, as both participants are actively participating in problem solving. To test this hypothesis, we calculated the average initiative shifts per line during KCC episodes and the average initiative shifts per line during problem solving outside of KCC episodes for each dyad. A paired t-test was then used to verify that there is a difference between the two groups. The t-test showed no significant difference in average dialogue initiative shifts in KCC episodes compared with non-KCC problem solving. However, there is a significant difference between average task initiative shifts in KCC episodes compared with the rest of the dialogue (t(57) = 3.32, p = 0.0016). The effect difference between the two groups (effect size = 0.65) shows that there is a meaningful increase in the number of task initiative shifts in KCC episodes compared with problem solving activity outside of the KCC episodes.

3.3 Indicators of task initiative shifts

Since our results show that task initiative shifts are conducive to learning, we want to endow our software agent with the ability to encourage a shift in initiative from the agent to the student, when the student is overly passive. The question is, what are natural indicators in dialogue that the partner should take the initiative? We explored two different methods for encouraging initiative shifts. One is that student uncertainty may lead to a shift in initiative. The other consists of cues for initiative shifts identified in related literature(Chu-Carroll and Brown, 1998; Walker and Whittaker, 1990).

Intuitively, uncertainty by a peer might lead to his partner taking the initiative. One possible identifier of student uncertainty is hedging. To validate this hypothesis, we annotated utterances in the corpus with hedging categories as identified in (Bhatt et al., 2004). Using these categories we were unable to reliably annotate for hedging. But, after collapsing the categories into a single binary value of hedging/not hedging we arrived at an acceptable agreement ($\kappa = 0.71$).

Another identifier of uncertainty is a student's request for feedback from his partner. When uncertain of his contribution, a student may request an evaluation from his peer. So, we annotated utterances with "request for feedback" and were able to arrive at an excellent level of intercoder reliability ($\kappa = 0.82$).

(Chu-Carroll and Brown, 1998) identifies cues that may contribute to the shift of task and dialogue initiative. Since task initiative shifts appear to identify KCC episodes, we chose to explore the following cues that potentially result in the shift of task initiative.

- Give up task. These are utterances where the student explicitly gives up the task using phrases like "Any other ideas?".
- Pause. A pause may suggest that the speaker has nothing more to say in the current turn and intends to give up his initiative.
- Prompts. A prompt is an utterance that has no propositional content.
- Invalid statements. These are incorrect statements made by a student.

Using hedging, request for feedback and initiative cues, we were able to identify 283 shifts in task initiative or approximately 67% of all task initiative shifts in the corpus. The remaining shifts were likely an explicit take over of initiative without a preceding predictor.

Since we found several possible ways to predict and encourage initiative shifts, the next step was to identify which of these predictors more often resulted in an initiative shift; and, for which predictors the resulting initiative shift more often led to an increase in the student's knowledge level. Table 4 shows the percentage of instances of each predictor that resulted in an initiative shift.

Cue/Identifier	Percent of instances that led to initiative shift
Hedge	23.94%
Request feedback	21.88%
Give-up task	20.00%
Pause	25.27%
Prompt	29.29%
Invalid statement	38.64%

Table 4: Cues for Shifts in Initiative

Along with the likelihood of a predictor leading to an initiative shift, we also examined the impact of a shift of task initiative on a student's level of knowledge, measured using knowledge score, calculated on the basis of the student model (see Section 4). This is an important characteristic since we want to encourage initiative shifts in an effort to increase learning. First, we analyzed initiative shifts to determine if they resulted in an increase in knowledge score. We found that in our corpus, an initiative shift leads to an increase in a student's knowledge level in 37.0% of task initiative shifts, a decrease in knowledge level in 5.2% of shifts and unchanged in 57.8% of shifts. Even though over one-half of the time knowledge scores were not impacted, in only a small minority of instances did a shift have a negative impact on a student's level of knowledge. Therefore, we more closely examined the predictors to see which more frequently led to an increase in student knowledge. The results of that analysis is show in table 5.

Predictor	Percent of shifts where knowledge level increased
Hedge	23.52%
Request feedback	17.65%
Give-up task	0.00%
Prompt	32.93%
Pause	14.22%
Invalid statement	23.53%

Table 5: Task Initiative Shifts/Knowledge Level Change

4 KSC-PaL, a software peer

Our peer-learning agent, KSC-PaL, has at its core the TuTalk System(Jordan et al., 2007), a dialogue management system that supports natural language dialogue in educational applications. Since TuTalk does not include an interface or a student model, we developed both in previous years. We also needed to extend the TuTalk planner to recognize and promote initiative shifts.

The user interface is structured similarly to the one used in data collection(see Figure 1). However, we added additional features to allow a student to effectively communicate with the KSC-PaL. First, all drawing and coding actions of the student are interpreted and passed to the agent as a natural language utterance. Graphical actions are matched to a set of known actions and when a student signals that he/she has finished drawing or coding either by ceding control of the graphical workspace or by starting to communicate through typed text, the interface will attempt to match what the student has drawn or coded with its database of known graphical actions. These graphical actions include not only correct ones but also anticipated misconceptions that were collected from the data collection interactions. The second enhancement to the interface is a spell corrector for "chat slang". We found in the corpus, that students often used abbreviations that are common to text messaging. These abbreviations are not recognized by the English language spell corrector in the TuTalk system, so a chat slang interpretation module was added.

KSC-PaL requires a student model to track the current state of problem solving as well as estimate the student's knowledge of concepts involved in solving the problem in order to guide its behavior. Our student model incorporates problem solution graphs (Conati et al., 2002). Solution graphs are Bayesian networks where each node represents either an action required to solve the problem, a concept required as part of problem solving or an anticipated misconception. A user's utterances and actions are then matched to these nodes. A knowledge score can be calculated at any point in time by taking a sum of the probabilities of all nodes in the graph, except the misconception nodes. The sum of the probabilities of the misconception nodes are subtracted from the total to arrive at a knowledge score. This score is then normalized by dividing it by the maximum possible knowledge score for the solution graph.

4.1 KSC-PaL and initiative

Since our corpus study showed that the level of task initiative can be used to identify when KCC and potentially learning is occurring, we have endowed KSC-PaL with behaviors to manipulate shifts in task initiative in order to encourage KCC and learning. This required three enhancements: first, the ability to recognize the initiative holder in each utterance or action; second, the ability to encourage the shift of initiative from the agent to the student; and three, extending the TuTalk planner so that it can process task initiative shifts.

As concerns the first step, that the agent recognize the initiative holder in each utterance or action, we resorted to machine learning. Using the Weka Toolkit(Witten and Frank, 2005), we explored various machine learning algorithms and feature sets that could reliably identify the holder of task initiative. We found that the relevant features of an action in the graphical workspace were substantially different from those of a natural language utterance. Therefore, we trained and tested separate classifiers for each type of student action. After examining a wide variety of machine learning algorithms we selected the following two classifiers: (1) K* (Cleary and Trigg, 1995), a clustering algorithm, for classifying natural language utterances which correctly classified 71.7699% of utterance and (2) JRip (Cohen, 1995), a rule-based algorithm, for classifying drawing and coding actions which correctly classified 86.971% of the instances.

As concerns the second step, encouraging initiative shifts so that the student assumes the task initiative, we use the results of our analysis of the indicators of task initiative shifts from Section 3.3. KSC-PaL will use prompts, request feedback and make invalid statements in order to encourage initiative shifts and promote learning.

Finally, we augmented the TuTalk planner so that it selects scripts to manage task initiative shifts. Two factors will determine whether a script that encourages initiative shifts will be selected: the current level of initiative shifts and the change in the student's knowledge score. Task initiative shifts will be tracked using the classifier described above. Scripts will be selected to encourage initiative shifts when the average level of initiative shifts is less than the mean initiative shifts in KCC episodes (calculated from the corpus data) and the student's knowledge level has not increased since the last time a script selection was requested. The scripts are based on the analysis of methods for encouraging initiative shifts described above. Specifically, KSC-PaL will encourage initiative shifts by responding to student input using prompts, requesting feedback from the student and encouraging student criticism by intentionally making errors in problem solving.

We are now poised to run user experiments. We will run subjects in two conditions with KSC-PaL: in the first condition (control), KSC-PaL will not encourage task initiative shifts and act more as a tutor; in the second condition, KSC-PaL will encourage task initiative shifts as we just discussed. One final note: because we do not want our experiments to be affected by the inability of the agent to interpret an utterance, given current NLU technology, the interface will "incorporate" a human interpreter. The interpreter will receive student utterances along with a list of possible matching concepts from TuTalk. The interpreter will select the most likely matching concept, thus assisting TuTalk in natural language interpretation. Note that the interpreter has a limited, predetermined sets of choices, corresponding to the concepts TuTalk knows about. In this way, his / her intervention is circumscribed.

5 Conclusions

After an extensive analysis of peer-learning interactions, we have found that task initiative shifts can be used to determine when students are engaged in knowledge co-construction. We have embedded this finding in a peer-learning agent, KSC-PaL, that varies its behavior to encourage initiative shifts and knowledge co-construction in order to promote learning. We are poised to run our user experiments, and we will have preliminary results available by the workshop time.

References

AA. VV. 2001. Computer Science, Final Report, The Joint Task Force on Computing Curricula. *IEEE Computer Society and Association for Computing Machinery, IEEE Computer Society.*

AA. VV. 2006. US bureau of labor statistics http://www.bls.gov/oco/oco20016.htm.

Khelan Bhatt, Martha Evens, and Shlomo Argamon. 2004. Hedged responses and expressions of affect in human/human and human computer tutorial interactions. In *Proceedings Cognitive Science*.

M. W. Birtz, J. Dixon, and T. F. McLaughlin. 1989. The effects of peer tutoring on mathematics performance: A recent review. *B. C. Journal of Special Education*, 13(1):17–33.

A. L. Brown and A. S. Palincsar, 1989. *Guided, cooperative learning and individual knowledge acquisition*, pages 307–226. Lawrence Erlbaum Associates, Hillsdale, NJ.

Jean Carletta. 1996. Assessing agreement on classification tasks: the kappa statistic. *Comput. Linguist.*, 22(2):249–254.

M.T.H. Chi, N. De Leeuw, M.H. Chiu, and C. LaVancher. 1994. Eliciting self-explanations improves understanding. *Cognitive Science*, 18(3):439–477.

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.

John G. Cleary and Leonard E. Trigg. 1995. K*: An instance-based learner using an entropic distance measure. In *Proc. of the 12th International Conference on Machine Learning*, pages 108–114.

P.A. Cohen, J.A. Kulik, and C.C. Kulik. 1982. Education outcomes of tutoring: A meta-analysis of findings. *American Education Research Journal*, 19(2):237–248.

William W. Cohen. 1995. Fast effective rule induction. In *Machine Learning: Proceedings of the Twelve International Conference*.

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.

Albert T. Corbett and John R. Anderson. 1990. The effect of feedback control on learning to program with the LISP tutor. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, pages 796–803.

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.

Davide Fossati, Barbara Di Eugenio, Christopher Brown, Stellan Ohlsson, David Cosejo, and Lin Chen. accepted for publication, 2009. Supporting Computer Science curriculum: Exploring and learning linked lists with iList. *EEE Transactions on Learning Technologies, Special Issue on Real-World Applications of Intelligent Tutoring Systems*.

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, Michelene 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 Conference 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, Brian Hall, Michael A. Ringenberg, Yui Cue, and Carolyn Penstein Rosé. 2007. Tools for authoring a dialogue agent that participates in learning studies. In *Artificial Intelligence in Education, AIED 2007*, pages 43–50.

S. Katz, J. Aronis, D. Allbritton, C. Wilson, and M.L. Soffa. 2003. Gender and race in predicting achievement in computer science. *Technology and Society Magazine, IEEE*, 22(3):20–27.

H. Chad Lane and Kurt VanLehn. 2003. Coached program planning: dialogue-based support for novice program design. *SIGCSE Bull.*, 35(1):148–152.

Karen E. Lochbaum and Candace L Sidner. 1990. Models of plans to support communication: An initial report. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 485–490. AAAI Press.

A. Mitrović, P. Suraweera, B. Martin, and A. Weerasinghe. 2004. DB-Suite: Experiences with Three Intelligent, Web-Based Database Tutors. *Journal of Interactive Learning Research*, 15(4):409–433.

R. Ploetzner, P. Dillenbourg, M. Preier, and D. Traum. 1999. Learning by explaining to oneself and to others. *Collaborative learning: Cognitive and computational approaches*, pages 103–121.

M. D. Rekrut. 1992. Teaching to learn: Cross-age tutoring to enhance strategy instruction. *American Education Research Association*.

Rod D. Roscoe and Michelene T. H. Chi. 2007. Understanding tutor learning: Knowledge-building and knowledge-telling in peer tutors' explanations and questions. *Review of Educational Research*, 77(4):534–574.

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.

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.

Kai Warendorf and Colin Tan. 1997. Adis-an animated data structure intelligent tutoring system or putting an interactive tutor on the www. In *Intelligent Educational Systems on the World Wide Web (Workshop Proceedings), at the Eight International Conference on Artficial Intellignece in Education*.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco.