Tutorial Dialogue as Adaptive Collaborative Learning Support

Rohit KUMAR, Carolyn P. ROSÉ, Yi-Chia WANG, Mahesh JOSHI, Allen ROBINSON
Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, USA
{rohitk,cp3a,yichiaw,maheshj,alr}@cs.cmu.edu

Abstract. In this paper we investigate the role of reflection in simulation based learning by manipulating two independent factors that each separately lead to significant learning effects, namely whether students worked alone or in pairs, and what type of support students were provided with. Our finding is that in our simulation based learning task, students learned significantly more when they worked in pairs than when they worked alone. Furthermore, dynamic support implemented with tutorial dialogue agents lead to significantly more learning than no support, while static support was not statistically distinguishable from either of the other two conditions. The largest effect size in comparison with the control condition was Pairs+Dynamic support, with an effect size of 1.24 standard deviations, where the control condition is individuals working alone with no support. Because the effect size achieved by combining the two treatments is greater than the effect achieved by either of the two treatments alone, we conjecture that each of these factors are contributing something different to student learning rather than being potential replacements for one another.

Keywords. Collaborative Learning, Dynamic Support, Tutorial Dialogue Agents

Introduction

In order to encourage productive patterns of collaborative discourse, researchers both in the Computer Supported Collaborative Learning (CSCL) tradition and the Educational Psychology tradition have separately developed approaches to using scripts for scaffolding the interactions between students, to help coordinate their communication, and to encourage deep thinking and reflection [1]. Script based support has traditionally taken on a wide range of forms including assignment of students to roles [2], provision of prompts during collaboration [3], design of structured interfaces including such things as buttons associated with typical "conversation openings"[4], instructions to guide learners to structure their collaboration [5], or even various forms of collaboration training [6]. Previous approaches to scripting have been static, one-size-fits-all approaches. In other words, the approaches were not responsive to what was happening in the collaboration. This non-adaptive approach can lead to over scripting [7] or interference between different types of scripts [8]. With these things in mind, and considering that ideally we would like students to internalize the principles encoded in the script based support, a more dynamic and potentially more desirable approach would be to trigger support based on observed need and to fade scaffolding over time as students acquire the skills needed to collaborate productively in a learning context. The concept of adaptive collaborative learning support has already been evaluated in a Wizard-of-Oz setup and found to be effective for supporting learning [9]. As a proof of concept of the feasibility and potential impact of such an approach, in this paper we describe an evaluation of a simple but fully-automatic approach to adaptive collaborative learning support.

Context sensitive or need based support necessitates on-line monitoring of collaborative learning processes. And, indeed, there has been much work in the computer supported collaborative learning community on modeling the process of collaborative learning with coding schemes applied to corpus data by hand [10,11,12]. Our own recent work towards automating the application of a well established collaborative learning process analysis coding scheme [13] demonstrates that patterns that indicate trouble in a collaborative discourse can be detected with a high degree of reliability and thus that a more dynamic support approach for collaborative learning is feasible [14]. The technology for automating such analyses is now publicly available[1]. So the time is ripe for exploring the use of this technology for triggering adaptive collaborative learning support.

In this paper we introduce and evaluate a new adaptive support mechanism specifically designed to draw out reflection using conversational agents that engage students in directed lines of reasoning [16]. This new collaborative learning support approach makes use of tutorial dialogue technology in the context of a simulation based learning task for college level thermodynamics instruction, building on previous results

[1] TagHelper tools can be downloaded from http://www.cs.cmu.edu/~cprose/TagHelper.html.

demonstrating the effectiveness of these tutorial dialogue agents for supporting learning of individuals working alone in this domain [17].

1. Architecture for Adaptive Collaborative Learning Support

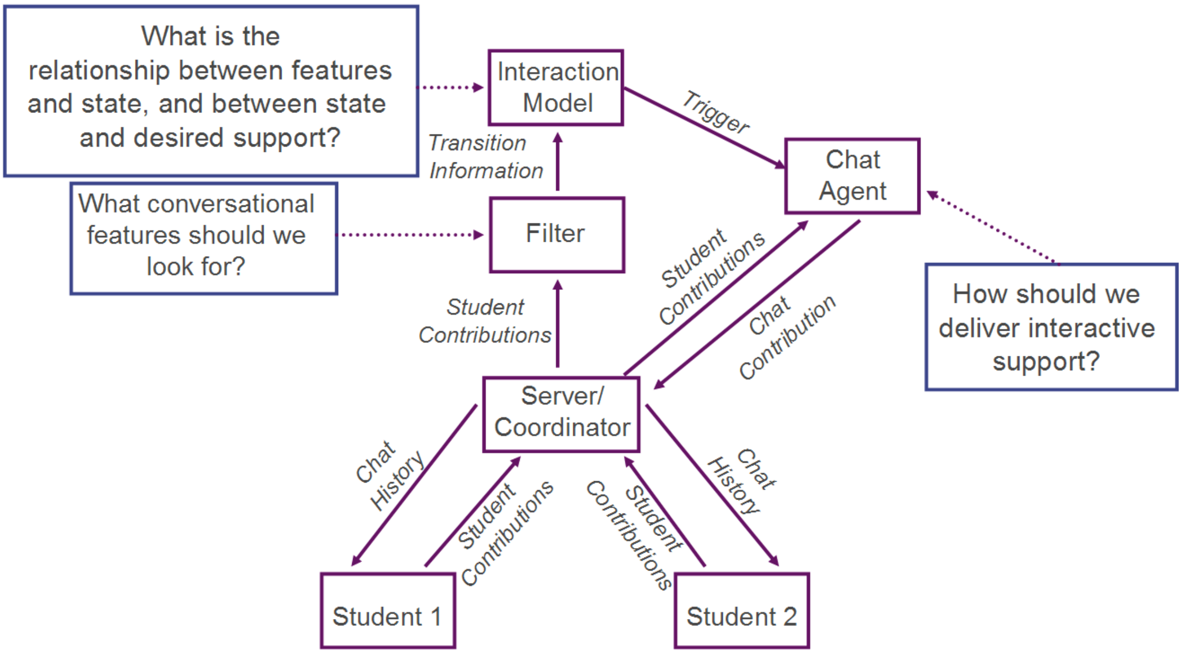

Figure 1 This figure displays our general architecture for enabling adaptive collaborative learning support. It can be customized in three ways: (1) the set of conversational features that the Filter is set up to identify in running conversations, (2) the way detected features are accumulated into a dialogue state as well as which states trigger support, and (3) the form of the support that gets triggered.

Figure 1 displays our general architecture for triggering adaptive collaborative learning support, which was used in the work reported in this paper. This architecture is meant to be general purpose. Here we describe the basic architecture and the three main ways it can be tailored to different learning contexts.

Each student has their own chat client, as displayed at the bottom of the diagram. Each chat client connects to a central Server/Coordinator that maintains a dialogue history with the contributions of all of the group members, including the contributions of the support agent. Thus, each student's client can display the entire interaction. As student contributions are accumulated in the history maintained by the Sever/Coordinator, these contributions are also passed through a Filter that is tailored to identify a pre-selected set of conversational events. As these events are detected, notification of these events is passed to the Interaction Model, which maintains the conversational state. Certain states trigger support from the Chat Agent.

The first way this architecture can be tailored is through the Filter. In the work reported in this paper, we use a simple topic oriented filter, which we describe in the next section. Thus, the conversational events it is meant to detect are topics that are related to the learning task. When students raise one of these topics, and it is detected with the Filter, notification is sent to the Interaction Model. However, we could have taken a much different approach. Rather than detecting which topics students discussed, we could have been detecting events associated with ways in which the students are interacting with one another, as in our previous work on automatic collaborative learning process analysis [14]. For example, we may have instead been examining the manner in which students were referring to the contributions of their learning partners, or monitoring whether students provided warrants to back up their claims.

The second means by which this architecture can be tailored is in the Interaction Module. In the current work, this module is implemented in a trivial manner. Each time a topic is detected that has not been previously detected within the conversation, it triggers support for reflection related to that topic. However, we could have taken an approach whereby we looked at how frequently within the past window of time students contributed a claim without an appropriate warrant. And each time that frequency dropped below a particular threshold, support related to warranting claims could be triggered. A third approach might be to wait until certain rare but important events occur, such as an indication that students have reached an impasse and are not able to proceed, and only then to trigger support from the Chat Agent.

The final manner in which this architecture can be tailored is the form of support offered to students by the Chat Agent. In our previous work we have evaluated support in the form of prompts requesting further elaboration on explanations [9] or prompts suggesting new categories of ideas to consider as part of an idea generation task [21]. In this study, support is offered in the form of directed lines of reasoning.

2. Experimental Infrastructure: The CycleTalk Chat Environment

We are conducting our research in the domain of thermodynamics, using as a foundation the CyclePad articulate simulator [15]. CyclePad was developed with the intention of allowing students to engage in design activities earlier in their engineering education than was possible previously. Our explorations of CyclePad use focus on design and optimization of thermodynamic cycles, specifically Rankine cycles. A thermodynamic cycle processes energy by transforming a working fluid within a system of networked components (condensers, turbines, pumps, and such). Power plants, engines, and refrigerators are all examples of thermodynamic cycles. Rankine cycles are a type of heat engine that forms the foundation for the design of the majority of steam based power plants that create the majority of the electricity used in the United States. There are three typical paradigms for design of Rankine cycles, namely the Simple Rankine Cycle, Rankine Cycle with Reheat, and Rankine Cycle with Regeneration. As students work with CyclePad on design and optimization of Rankine Cycles, they start with these basic ideas and combine them into novel designs.

2.1. The Original CycleTalk System

In our previous work, we have integrated the CyclePad simulation environment with example tracing tutoring systems [18] using the Cognitive Tutor Authoring Tools [19]. We have evaluated this integrated version in our previous studies [17,20]. The integration with example tracing tutor technology allows us to offer students the opportunity to ask for hints as they work with the environment. In our previous studies [9], we have demonstrated that integrating tutorial dialogue agents with this environment significantly improves student learning. The integration of CyclePad with hints provided by example tracing tutors and tutorial dialogues provided by the TuTalk tutorial dialogue engine [16] comprises what we refer to as the original CycleTalk system. In our previous work, we have demonstrated that students learn significantly more from CycleTalk when dialogues are provided in addition to hints rather than only hints. One hypothesis about the reason for the effectiveness of adding tutorial dialogue to this environment is that students do not necessarily take advantage of all of the opportunities they have for productive reflection as they interact with a simulation environment like CyclePad. Thus, in the current study we focus specifically on this reflective function of dialogue by separating the interaction with the simulation from the interaction with the dialogue agents into two distinct stages, where the interaction with the dialogue agents occurs strictly after the formal instruction that we evaluated previously.

2.2. The CycleTalk Chat Environment

In the current study, students complete all of the instruction we used in our previous studies before they even take the pretest. However, the tutorial dialogue agents that were integrated with the CycleTalk environment previously are now used as part of the collaborative learning support we want to test as an intervention. Thus, although we still use the CycleTalk system during the instruction that occurs before the pretest, we use a version of the system without the tutorial dialogue component while students are working through those materials. Further details about our experimental design are explained in Section 3. In the remainder of Section 2.2 we describe how we tailored our general architecture for adaptive collaborative learning support for use in the current study. As described in the previous section, we do this by tailoring the Filter, the Interaction Model, and the Chat Agent.

2.2.1. The Filter

In the previous section we described the role of the Filter in our architecture for adaptive collaborative learning support. In the CycleTalk Chat Environment, the Filter was constructed semi-automatically from a topic segmented and labeled corpus collected in a previous CycleTalk study involving human tutors [20]. That corpus is comprised of 379 topic segments belonging to 16 topics. The first step of our process was to identify the terms that were strongly associated with each of these topics. We did this using a standard metric used in the information retrieval community referred to as term frequency inverse document frequency or TF-IDF. Using this metric, words that occur frequently within a document but relatively infrequently across documents receive a high weight for that document, indicating that term is very informative about what is special about that document. In order to compute these term weights for each topic, we combined all segments associated with the same topic into a single "document". We then preprocessed each document by stripping off the inflectional endings on words using a publicly available stemmer in order to generalize across different forms of the same word. We then removed stop words, which are function words carrying no content such as determiners and prepositions, and then extracted a list of unigrams (i.e., single words) and bigrams (i.e., pairs of words occurring next to one another). Furthermore, numbers were replaced with tokens representing ranges of numbers that are meaningful in the domain. For example, 2000-25000 is an appropriate range for maximum pressure. For each unigram and bigram, we then computed a TF-IDF weight for each document. The result of this process was a weight associated for each phrase per topic. Using these weights, a topic can be selected by averaging the term weights for all unigrams and bigrams identified in a student contribution for each topic, and selecting the topic that receives the highest weight. Topics not associated with any of the 7 knowledge construction dialogues used in this study were ignored. Furthermore, only topics receiving an average weight of at least 2.5 were considered. The 2.5 threshold was determined experimentally over the annotated development data.

Based on an error analysis of the performance of this filter, we determined that the biggest source of errors was on false positives, or identifying a spurious topic where it was not raised. Thus, we implemented an additional filter that attempts to distinguish important contributions where a topic is raised from those where no topic is raised. We used TagHelper tools [14] to construct a support vector machine (SVM) model based on our topic labeling. Thus, the final topic filter first uses this model to determine whether a topic may have been raised, and then used the filter described above to determine which topic it is most likely to be.

2.2.2. The Interaction Model

The Filter just described identifies when concepts are raised that are related to one of the 7 directed lines of reasoning related to the concepts we want to stimulate reflection about. Whenever a concept is identified in the student input, a trigger is sent to the Chat Agent to launch the corresponding knowledge construction dialogue. Beyond this, the Interaction Model also includes a collaboration policy that aims to maintain the students' engagement and to motivate the students to participate equally in the conversation. First, the student(s) are prompted once every minute with a hint about a concept that has not been discussed in the conversation so far. The prompts are addressed to a specific student. In case multiple students are involved in the conversation, the student contributing fewer concepts is more likely to be prompted. Students are also prompted at the end of every knowledge construction dialogue, to indicate the end of a guided interaction and to encourage the students to get back into idea generation and information gathering. These prompts are generic in nature and do not hint towards any particular concept.

2.2.3. The Chat Agent

The Chat Agent delivers two types of support. The first form of support is the hints triggered by the collaboration policy described above. The form of the hints was based on a design tested in previous collaborative idea generation studies [21,22]. These prompts are meant to stimulate thinking towards an idea without giving the idea away. One example meant to elicit the idea of manipulating the maximum pressure of the cycle is “Try to think of an idea related to manipulating a property of the pump.”

The second form of support is the directed lines of reasoning implemented with the TuTalk dialogue engine. We use a subset of the same dialogue agents that were developed and evaluated in our previous work [9]. In the context of the current study, the purpose of these directed lines of reasoning is to support the student in reflecting on how concepts from the instructional phase can be applied in the exploratory phase. Each of 7 directed lines of reasoning focus on one or two key relationships between cycle variables, for example how the maximum temperature of the cycle affects the thermal efficiency of the cycle. These directed lines of reasoning lead the student to reflect on the relationships between the variables and what implications that relationship should have on design and optimization of thermodynamic cycles. One of the interventions we test in our study is the use of conversational agents as a form of dynamic learning support. In the dynamic support conditions, the CycleTalk tutorial dialogue agent is a participant in the conversation. In addition to the typical knowledge construction dialogues provided by the CycleTalk agent, a collaboration policy agent monitors the chat behavior and inserts hints and prompts to encourage the conversation to proceed in a productive direction.

2.3. An Example Interaction

Here is an example interaction involving two students and the CycleTalk agent:

ST1: well what valtues do u have for the reheat cycle ?
Tutor: Let's think about the motivation for Reheat. What process does the steam undergo in the Turbines ?
ST2: for some reason I said temperature at turbine to be like 400 C

Tutor: Steam undergoes Isentropic Expansion in a turbines that converts the heat energy of the steam into work by rotating the blades. What happens to the Quality of steam during Expansion ?
ST1: was that the max eta thermal value?
ST2: shuoldnt quality stay the same through the turbine
Tutor: As steam expands in a turbine, its quality degrades as the moisture content in the steam increases. Should we use steam of degraded quality in the turbines ?
ST1: ya
Tutor: We should NOT use low quality steam in turbines as it condenses on the turbine blades and damages the turbine running at high speeds Does this constrain the amount of work steam can do in turbines ?
ST1: oh. no.
Tutor: Since quality of steam falls gradually in the turbines, we design turbines such that the quality remains above acceptable levels [0.85]. …

As the students are chatting, eventually Student 1 utters a contribution that triggers one of the concepts the Filter is looking for, namely the idea that Reheat is a form of Rankine Cycle that has the potential to achieve a higher efficiency than a Basic Rankine Cycle. So in the next turn, the Chat Agent, labeled here as Tutor, initiates a reflective dialogue about the concept of Reheat cycles. Initially, the students seem to ignore the dialogue. Student 2 responds to Student 1's contribution. The tutor treats this as a demonstration of a lack of knowledge and proceeds to offer instruction to lead the student to an answer to its ignored question. Student 1 then responds to Student 2, again seemingly ignoring the tutor's question, but then Student 1 jumps in with an answer. After Student 1 provides a serious response to one of the Tutor's questions, then Student 1 begins to interact with that agent.

3. Method

3.1. Materials

The domain specific materials used in the study, which consisted of training materials for the simulation environment, pre/post test, and a 31 page booklet including introductory reading material about rankine cycles and focused readings with suggested illustrative analyses to perform using the CyclePad simulator for three forms of rankine cycles, were all developed by a Carnegie Mellon University mechanical engineering professor with the help of three of his graduate students and input from our team. These domain specific materials were exactly the same across conditions. Furthermore, all of the content presented in the set of dialogues the environment is capable of conducting with students in the Dynamic Support conditions is included in the reading materials in the form of "targeted text minilessons" [23], which present the same information using virtually identical phrasing, except that they are not interactive. The questionnaire, developed in our prior work [24] and administered only to students working in the Pairs conditions, consisted of 8 questions meant mainly to evaluate how positively or negatively students felt about their collaboration experience.

3.2. Experimental procedure common to all condition

The experimental procedure consisted of 8 steps. (1) At the beginning of the lab session, students were lead through formal training on the simulation software from an instructor using power point slides and the simulation environment. (15 minutes). (2) The students then worked through the 32 page booklet, which presented to them the formal domain content instruction and instructed them to do specific activities with the simulation environment (70 minutes). (3) Subsequent to the formal instruction, but immediately prior to the experimental manipulation, students took the pretest (15 minutes).

Steps (4)-(6) together constituted an exploratory exercise developed to allow students to apply the knowledge they learned during step (2). Immediately prior to step (4) of the experimental procedure, students were given instructions for steps (4)-(6) that described the terms of a "contest" to use what they just learned to design, build, and evaluate two rankine cycle designs with the best thermal efficiency they can achieve. A prize of a $25 gift certificate was promised for the student who would achieve the highest average thermal efficiency between the two designs. Second prize was a $10 gift certificate. For step (4), students were instructed to review what they learned and insert their notes into an MSN-messenger like chat buffer. The experimental manipulation took place during phase (4). Step (4) lasted for 25 minutes. (5) Next was the Planning/Synthesis step where students were instructed to review the notes generated during Step (4), and to form 2 concrete plans (10 minutes). (6) Finally, during the Implementation/Evaluation step, students were instructed to build their designs in the simulation environment and evaluate their thermal efficiency (25 minutes).

(7) In order to assess what students learned from the exploratory exercise, they then took a post-test (15 minutes). (8) In the Pairs conditions only, the students then took the questionnaire.

3.3. Experimental design

Our experimental manipulation consisted of 6 conditions resulting from a 3X2 full factorial design crossing a 3 level support factor (no support, static support, or dynamic support) with a two level collaboration status factor (working with a partner or working alone). The experimental manipulation took place during step (4) only, where students typed notes, ideas, and thoughts into an MSN like chat buffer as described above. In all other phases students worked independently in a consistent fashion across conditions. The instructions given to students in all conditions were identical except for the manipulation specific instructions inserted at the end of the instructions for step (4), which are described below. All students were instructed to review specific relevant portions of the book they had all been given, and which they had worked through during step (2) of the experimental procedure. Thus, the experimental manipulation only affected the manner in which students interacted with the chat buffer, i.e., whether that interaction included interaction with another student and/or interaction with a dialogue agent offering dynamic support. More details are offered in the next section

3.4. Outcome Measures

We looked at three outcome measures of instructional effectiveness. Two outcome measures were assessed by means of a Pre/Post test. 42 multiple choice and short answer questions were used to test analytical knowledge of Rankine cycles, including relationships between cycle parameters. An important aspect of this was a set of prediction questions where students were told to predict the impact of a specific change in one cycle parameter on several other cycle parameters. The other part of the test was a set of 8 open response questions assessing conceptual understanding of Rankine cycles. The third type of outcome measure we looked at was ability to apply knowledge to build and optimize a Rankine cycle using CyclePad during a phases (4)-(6) of the experimental procedure. This was assessed by examining the simulator log files the students saved from the Implementation/Evaluation step (i.e., step 6) of the experimental procedure.

3.5. Participants

We conducted our study over a four day period of time as part of a sophomore Thermodynamics course at Carnegie Mellon University. The week before the study, students were introduced to the topic of Rankine cycles during the lecture part of their course. They then completed a take home assignment related to Rankine cycles. Finally, they participated in a 3 hour lab session in which the experiment took place. Students were randomly assigned to conditions, subject to scheduling constraints. Each of 6 lab sessions were assigned to one of the 6 experimental conditions in such a way that we balanced as much as possible the distribution of experimental configurations across days. 87 students participated in the study, including 13 students in each of the conditions where students worked alone, and 16 in each condition where students worked with a partner.

4. Results

4.1. Learning Gains Analysis

The clearest indication of the complementary value of collaboration and tutorial dialogue support come from the multiple choice portion of the test. In our analysis, we used an ANCOVA model with our two independent variables, where post test score was the dependent variable, and the pretest score was the covariate. We found a significant main effect in favor of the Pair condition $F(1,80) = 4.64$, $p < .05$, effect size .4 standard deviations. Furthermore, we found a significant main effect of the support variable $F(2,80) = 5.7$, $p < .005$, effect size .7 standard deviations. A pairwise Tukey test post-hoc analysis revealed that only the contrast between No Support and Dynamic Support was significant, with Dynamic Support being preferred. There was also a marginal interaction effect between the two independent variables $F(2,80) = 2.7$, $p = .07$. An analysis of the relative effect sizes of the experimental conditions in comparison with the Individual-No Support condition reveals that the Pairs-Dynamic Support condition achieved the highest effect size, namely 1.24 standard deviations. Pairs with Static Support achieved an effect size of .9 standard deviations over the control condition. Individuals with Dynamic Support achieved an effect size of 1.06.

The conceptual portion of the test revealed an advantage for tutorial dialogue based support but not collaboration. As before, we used an ANCOVA model with our two independent variables, where conceptual post test score was the dependent variable, and the conceptual pretest score was the covariate. However, here we found only a marginal main effect of the support variable with the full ANCOVA model. With a simpler ANCOVA model where we drop the Pair versus Individual factor and only evaluate the effect of the Support variable we find a significant main effect of the support variable $F(2,83) = 3.07$, $p = .05$. A pairwise Tukey test post-hoc analysis revealed that only the contrast between No Support and Dynamic Support was significant, with Dynamic Support being preferred. The effect size is .5 standard deviations. Based on an informal analysis of the chat logs, it is not surprising that we only see an effect of collaboration on the multiple choice portion of the test since the majority of the conversations focused on content covered in this portion of the test, specifically relationships between cycle parameters.

The practical assessment revealed no differences between conditions. We examined student success on the Execution/Evaluation stage. 91% of students were successful at building one fully defined cycle, and 64% of students were successfully able to define two. There was no significant effect of condition on success rate or average efficiency of best performing cycle.

4.2. Questionnaire

The questionnaire data in some ways corroborated the learning gains analysis but also alerted us to some issues that we must address in our ongoing work. We examined the results of the questionnaire using a repeated measures logistic regression model with Question Type (i.e., Knowledge, Help, Engagement, and Benefit) as a repeated measure, with each level corresponding to the subset of the questionnaire related to these specific aspects of the collaboration. Since students assigned a value between -5 and +5 as their answer to each question, we assigned a 1 to a student for a question category type if their average score across questions related to that category was above 0, and 0 otherwise. Additionally, we included the Support condition variable as well as Student nested within Support condition in the model. The R^2 value of the model was .35. There was no main effect of Support condition, although there was a main effect of Student, Question Type, and the interaction between Question Type and Support Condition (LR-Chi2 = 17.1, $p < .01$). Consistent with what would be expected based on the test results, students in the Dynamic Support condition rated themselves as benefiting significantly more from the collaboration. However, contrary to expectation, students in the Dynamic Support condition rated themselves and their partner as significantly less engaged in the material during the collaborative portion of the experimental manipulation than students in the other two support conditions. While the results on the test show promise that our Dynamic support mechanism can lead to learning benefits, consistent with the questionnaire data we noticed many comments in the chat logs that indicated that students were frustrated with the dialogue agent. An informal analysis of the chat logs reveals that the dialogue agents sometimes interfered with the conversation between students in the Pairs-Dynamic Support condition, and that the students seemed frequently to ignore what the dialogue agents were saying, interacting with each other around the agent rather than interacting with the agent.

5. Conclusions

We have investigated the role of reflection in simulation based learning by manipulating two independent factors that each separately lead to significant learning effects. The important finding is that because the effect size achieved by combining the two treatments is greater than the effect achieved by either of the two treatments, we conjecture that each of these factors must contribute something different to student learning rather than being replacements for one another. Consistent with this, an informal analysis of the chat logs reveals that the nature of the reflection students engage in with each other is distinct from that in connection with the dialogue agents. Specifically, while the dialogue agents lead students crisply through a reasoning process that connects observations with principles and ultimately with design decisions, student discussions were more intersubjective in nature. We found students negotiating over differences in findings and interpretations of findings and confirming or modifying their partial understandings and inferences. Equally important, an analysis of the chat logs and questionnaire data demonstrate that there is still room for significant improvement of our design for adaptive collaborative learning support. In our current work we are continuing to investigate the reasons for student frustration with the dialogue agents in order to develop a more effective approach for future studies.

Acknowledgements

This project is supported by ONR Cognitive and Neural Sciences Division, Grant numbers N000140410107 and N000140510043.

References

[1] Kollar, I., Fischer, F. & Hesse, F. W. (2006). Computer-supported cooperation scripts - a conceptual analysis. Educational Psychology Review.

[2] Strijbos, J. W. (2004). The effect of roles on computer supported collaborative learning, Open Universiteit Nederland, Heerlen, The Netherlands

[3] Weinberger, A. (2003). *Scripts for Computer-Supported Collaborative Learning Effects of social and epistemic cooperation scripts on collaborative knowledge construction, Unpublished* Dissertation, University of Munich.

[4] Baker, M., & Lund, K. (1997). Promoting reflective interactions in a CSCL environment. *Journal of Computer Assisted Learning,* 13, 175-193.

[5] Webb, N. & Farivar, S. (1999). Developing Productive Group Interaction, in O'Donnell & King (Eds.) *Cognitive Perspectives on Peer Learning*, Lawrence Erlbaum Associates: New Jersey.

[6] Rummel, N., Spada, H. & Hauser, S. (2006). Learning to collaborate in a computer-mediated setting: Observing a model beats learning from being scripted. In S. Barab, D. Hickey & K. Hay (Eds.) *Proceedings of the Seventh International Conference of the Learning Sciences*. Mahwah, NJ: Lawrence Erlbaum Associates

[7] O 'Donnell, A. M. Structuring Dyadic Interaction Through Scripted Cooperation , in O'Donnell & King (Eds.) Cognitive Perspectives on Peer Learning, Lawrence Erlbaum Associates: new Jersey, 1999.

[8] Weinberger, A., Ertl, B., Fischer, F., Mandl, H. Cooperation scripts for learning via web-based discussion boards and videoconferencing. In Proc. The first joint meeting of the EARLI SIGs "Instructional Design" and "Learning and Instruction with Computers" Knowledge Media Research Center (2004), 22-28.

[9] Gweon, G., Rosé, C. P., Zaiss, Z., & Carey, R. (2006). Providing Support for Adaptive Scripting in an On-Line Collaborative Learning Environment, Proceedings of CHI 06: ACM conference on human factors in computer systems. New York: ACM Press.

[10] De Wever, B., Schellens T., Valcke, M., & Van Keer H. (2006). Content analysis schemes to analyze transcripts of online asynchronous discussion groups: A review. *Computers and Education*, 46, 6-28.

[11] van der Pol, J., Admiraal, W. & Simons, P. R. J. (2006) The affordance of anchored discussion for the collaborative processing of academic texts. *International Journal of Computer-Supported Collaborative Learning, 1 (3)*

[12] Webb, N. M. (1993). Collaborative group versus individual assessment in mathematics: Processes and outcomes. *Educational Assessment*, 1, 131-152.

[13] Weinberger, A., & Fischer, F. (2006). A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. *Computers & Education*, 46, 71-95.

[14] Donmez, P., Rose, C. P., Stegmann, K., Weinberger, A., and Fischer, F. (2005). Supporting CSCL with Automatic Corpus Analysis Technology, in Proceedings of Computer Supported Collaborative Learning.

[15] Forbus, K. D., Whalley, P. B., Evrett, J. O., Ureel, L., Brokowski, M., Baher, J., Kuehne, S. E. (1999). CyclePad: An Articulate Virtual Laboratory for Engineering Thermodynamics. Artificial Intelligence 114(1-2): 297-347.

[16] Jordan, P., Ringenberg, M. and Hall, B. (2006). Rapidly Developing Dialogue Systems that Support Learning Studies. In Proceedings of ITS06 Workshop on Teaching with Robots, Agents, and NLP.

[17] Kumar, R., Rosé, C. P., Aleven, V., Iglesias, A., Robinson, A. (2006). Evaluating the Effectiveness of Tutorial Dialogue Instruction in an Exploratory Learning Context, Proceedings of the Intelligent Tutoring Systems Conference.

[18] Aleven, V, & Rosé, C. P. (2005). Authoring plug-in tutor agents by demonstration: Rapid rapid tutor development, Proceedings of AI in Education '05.

[19] Koedinger, K. R., Aleven, V., Heffernan, N., McLaren, B., & Hockenberry, M. (2004). Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration. In J. C. Lester, R. M. Vicario, & F. Paraguaçu (Eds.), Proceedings of Seventh International Conference on Intelligent Tutoring Systems, ITS 2004 (pp. 162-174). Berlin: Springer Verlag.

[20] Rosé, C. P., Aleven, V., Carey, R., Robinson, A., Wu, C. (2005). A First Evaluation of the Instructional Value of Negotiatble Problem Solving Goals on the Exploratory Learning Continuum, Proceedings of AI in Eduction '05

[21] Wang, H. C., Rosé, C. P., Cui, Y., Chang, C. Y., Huang, C. C., & Li, T. Y. (2007). Thinking Hard Together: The Long and Short of Collaborative Idea Generation in Scientific Inquiry, Proceedings of CSCL 2007

[22] Nijstad, B. A., & Stroebe, W. (2006). How the group affects the mind: a cognitive model of idea generation in groups. Personality and Social Psychology Review, 10(3), 186-213.

[23] VanLehn, K., Graesser, A., Jackson, G. T., Jordan, P., Olney, A., Rosé, C. P., (in press). Natural Language Tutoring: A comparison of human tutors, computer tutors, and text. Cognitive Science.

[24] Gweon, G., Rosé, C. P., Albright, E., & Cui, Y. (2007). Evaluating the Effect of Feedback from a CSCL Problem Solving Environment on Learning, Interaction, and Perceived Interdependence, Proceedings of CSCL 2007.

.