Collaborative Intelligent Tutoring Systems: Comparing Learner Outcomes Across Varying Collaboration Feedback Strategies

Rachel Harsley, University of Illinois at Chicago, rharsl2@uic.edu
Barbara Di Eugenio, University of Illinois at Chicago, bdieugen@uic.edu
Nick Green, University of Illinois at Chicago, ngreen21@uic.edu
Davide Fossati, Emory University, davide@fossati.us

Abstract: In this paper, we present a collaborative extension of our ITS for Computer Science (CS) Education. The design of the collaborative version was motivated by noted benefits of collaborative learning including heightened retention of underrepresented students, particularly as demonstrated through pair programming within the CS domain. In this paper, we examine the outcome of two designs of the collaborative system with varying degrees of collaboration feedback. In the *unstructured* version, pairs are presented with no collaboration feedback while in the *semistructured* version of the system, pairs are given visual feedback in regards to their group and individual performance. We collected log data of system use as well as audio recordings of pairs. We found that students in both conditions experienced significant learning gains. Shifts in dialogue initiative where significantly positively correlated to learning gains in both conditions. However, students provided with additional collaboration feedback, exhibited less planning and overall symmetry.

Introduction

This study offers a comparative analysis on the effect of collaboration feedback within a collaborative tutoring system for Computer Science (CS) Education. The work synthesizes findings from the research domains of CSCL, ITS, and CS Education, and thus offers a foundational perspective on a growing area of Collaborative Intelligent Tutoring Systems (CITs). The study aims to provide insights on how pairs respond to performance feedback as well as the effect of tutor collaboration structuring on planning, symmetry, and learning gain.

While Intelligent Tutoring Systems (ITS) have offered a viable solution to the issues of automation and scalability, they have also been traditionally geared toward one-on-one, student-tutor. However, the growing model of CS Education has shifted to an emphasis on collaborative work and has resulted in higher retention rates of underrepresented students and better learning (Porter et al., 2013). Specifically, CS educators have adopted the practice of pair-programming for the classroom (Porter et al., 2013; Salleh et al., 2011). Our collaborative tutoring system, Collab-ChiQat Tutor, situates students as pair-programmers as they work to solve coding problems with the aid of the computer tutor. With respect to the spectrum of means to structure collaborative activity, we designed two versions of the system offer differing levels of collaboration feedback (Harsley et al., 2016).The *semistructured* system provides visual feedback on group and individual performance while the unstructured system does not. This study outlines the results of a comparative study of system use in an introductory undergraduate CS course and answers the question of how feedback structuring affect learning and collaborative interaction.

Motivating work

Though the historic focus of ITS development has been toward one-on-one tutoring, in more recent years, several one-on-one tutoring systems have been extended to support collaborative learning (Magnisalis, Demetriadis, & Karakostas, 2011; Olsen, Aleven, & Rummel, 2015). ITS researchers and developers are motivated by the noted benefits of collaborative learning as documented in CSCL literature. These include learning for transfer and learning gains that exceed the best of individual learners (Kaptelinin, 1999). However, it is well accepted that effective collaboration and student learning does not follow simply by placing students in groups. Instead, broadly, much CSCL research has examined how collaborative activities can be designed and structured in order to facilitate the most desirable outcomes. One such method is feedback via the display of group and individual performance. Individual and group participation visualization, peer feedback visualization, and the overall symmetry of participation have led to higher signs of engagement and improved performance (Janssen et al., 2007; Phielix et al., 2011).

One of Collab-ChiQat Tutor's primary goal is to help battle the problem of student retention that has plagued the CS discipline (Porter et al., 2013). This issue especially effects underrepresented students in the discipline including women and minorities (Washington et al., 2015). Collab-ChiQat Tutor's modules provide tutoring for CS data structures and algorithms which are significant hurdles for students that also impede retention efforts (Green et al., 2015).

Undoubtedly, both the CSCL and ITS community shape the current context of technology-enhanced learning. CSCL shows that students learn effectively in groups given proper mechanisms and activities. Moreover, it has established that computational environments can support the structuring of this collaboration. On the other hand, ITSs offer adaptive learner support that models an individual user, the learning domain, and the tutoring strategy. We reconceptualized our tutoring system with these bodies of research as foundation along with the CS model of pair programming. Notably, we recognize that the role of the tutor in structuring collaboration can widely range from limited structure with no collaboration feedback to high structuring with role definitions, group formation, and even timing of communication (Harsley, 2014). Thus, this study contrasts two methods of tutor collaboration structuring.

Methods

The *unstructured* interface consisted of four components; 1) textual problem 2) graphical problem representation 3) tutor feedback area and 4) coding interface. The *semistructured* version added a fifth panel, the collaboration panel, which provided graphical representation of the pair's individual and group performance (see Figure 1). The design of the collaboration panel was intended to promote self and group reflection, role-switching, and knowledge generative activities. We have reported more on the system architecture in prior work (Harsley et al., 2016). Seven problems were presented to students in increasing order of difficulty.

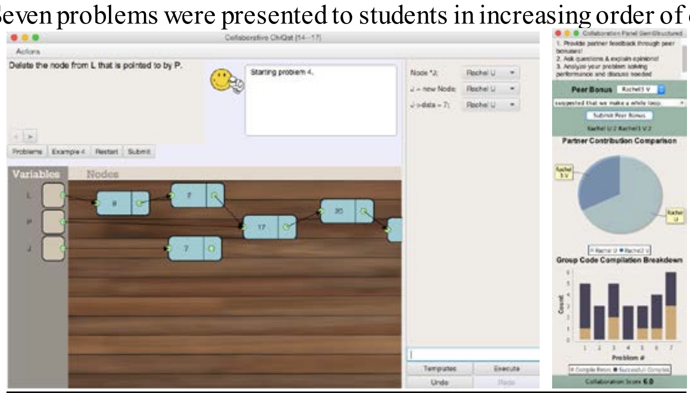

Figure 1. Main tutoring system interface (left) and *semistructured* condition collaboration panel (right).

This study was conducted in an introductory computer science course for undergraduates at a large public university. It consisted of a one-time intervention in which student used the tutoring system during a single lab session. 21 pairs participated in the study, with 41 students electing to share their data. Students chose their own partners and pairs were randomly assigned to a condition. In both conditions, students individually completed timed, identical tests. The test covered key concepts regarding the tutoring topic of CS linked lists. After completion of the post-test, students also took a brief survey regarding their overall experience with the system and general dispositions in regards to working with pairs. In both cases, students used the system for a total of 40 minutes. Seven coding problems were available for students to solve sequentially.

The tutoring system exhaustively logged student interaction with the systems. This included time-stamped traces of student clicks, keyboard events, the time to start a problem, the number of undo operations, and even number of lines coded before switching *driver,* and tutor feedback allowing later recreation of the students' activity on a fine grained level. Moreover, we collected audio recordings of each pair as they worked with the system. The system used a real-time estimate of student dialogue user automatic speech recognition. However, the recorded audio data was later transcribed manually post-intervention. We generated transcription-based features including counts of domain-related words and number of utterances. Moreover, we automatically labelled each transcribed utterance as either 1)question 2)command 3)prompt or 4)assertion following Walker and Whitaker's utterance based control rules (Walker & Whittaker, 1990). These labels were then used to track shifts in linguistically-based initiative. Initiative occurs when a speaker contributes new content (including questions) to the conversation that are not in response to the other participant.

After establishing our feature set, we used multiple linear regression to model post-test scores. A model was created for each feature along with a student's pre-test score as a co-variate. Our goal in this analysis was to establish which features were significant correlates to student learning. We followed the linear regression analysis with unpaired t-tests between every feature as compared between conditions. For example, we compared the time to start problem one in the *unstructured* condition versus the *semistructured* condition. This analysis would allow us to establish how different methods of structuring collaboration guidance affect collaboration and overall system interaction. Finally, we examined the student responses to survey questions to see differences in perception of collaboration.

Findings

Learning

We begin our contrastive analysis by comparing the learning gains of students measured as the difference between pre and post test scores. In both conditions we found a significant difference in learning gain ($p<.01$) establishing that students learned from using Collab-ChiQat Tutor. Further, there was not a significant difference in pre-test scores or learning gains between conditions. The learning gains are shown in Table 1.

Table 1: Student learning gains.

Condition	N	Pre-test		Post-test		Gain	
		μ	σ	μ	σ	μ	σ
Unstructured	22	.46	.20	.58	.21	.12	.19
Semistructured	19	.57	.23	.67	.24	.11	.28

Symmetry, planning, and pair programming perception

After establishing the effect of both conditions on student learning, linear regression analysis revealed that both symmetry was a significant correlate to learning. Namely, the number of times a student took dialogue initiative was significantly positively correlated to learning in both conditions ($p < .05$). Intuitively, as students work to solve the problem together, the initiative should shift between students. However, in the *semistructured* condition, the amount of times a partner took initiative was negatively significantly correlated to learning. This implies that the *semistructured* condition did not promote a balanced, or symmetric relationship. As expected, learning occurred as a student took initiative, however, as their partner took initiative, learning suffered. This finding was confirmed with analysis of student turn taking behavior while writing code. In the *semistructured* condition, the difference in coding turns between partners is significantly positively correlated to learning ($p = 0.04$, Adjusted-$R^2 = 0.73$). This means that as students had a large difference in coding taking turns, or one student dominated, learning improved.

Features pertaining to planning also played a key role in modelling learning. The time to start a problem is the time spent between when the problem is introduced and when students submit their first line of code. The time to start problems four through seven was positively significantly correlated to learning in the unstructured condition ($p < .05$). The importance of time before coding is intuitive as well as its approximation for planning time. Moreover, given our audio analysis, this time was most often spent discussing the meaning of the problem and proposing a solution approach. Notably, there was also a significant difference between *unstructured* and *semistructured* conditions in the time to start problems four and five ($p < .05$). As the problems increased in difficulty, the majority of students did not complete problems six and seven, thus we find no significant difference between conditions in these problems. However, these findings suggest that, unlike the students in the *unstructured* condition, students in the *semistructured* condition did not engage in planning as problem difficulty increased.

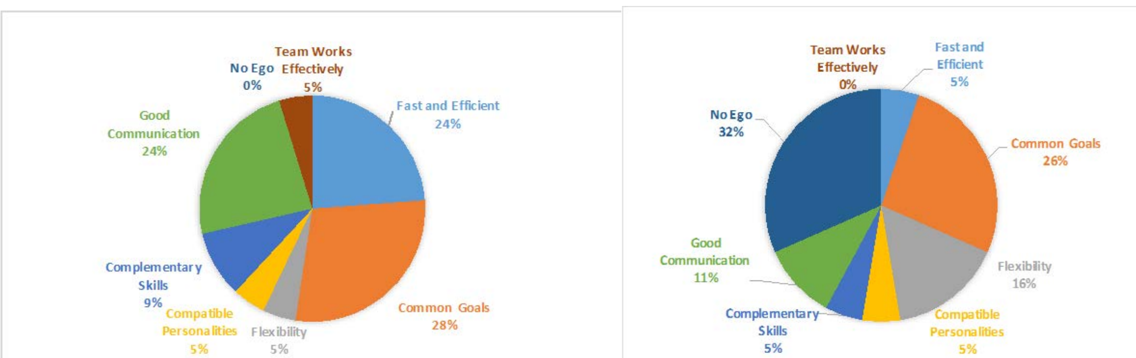

Figure 2. *Unstructured* (left) and *semistructured* (right) most important attribute of a pair programming team.

Lastly, survey results from students revealed that roughly the same proportion of students found the system helpful and interesting across conditions. However, students in each condition had distinctly differing perspectives on the attributes of a good pair programming team. Students in the *unstructured* condition ranked "Common Goals", "Good Communication", and "Fast and Efficient" to be the top three attributes. Contrastingly, students in the *semistructured* condition ranked "No Ego", "Common Goals", and "Flexibility" as top attributes. Notably, the *semistructured* attributes place more importance on individual acts that may be perceived as deference to their partner. On the other hand, *unstructured* users focus on group traits. A visualization of student responses is given in Figure 2. This same trend persists in student's free response to top attributes. Students in the *semistructured* condition used words such as "understanding" and "confidence" which do not appear at all or as frequently in *unstructured* student responses. Instead, the higher frequency of *unstructured* responses such as "respect" and "trust" allude to more group symmetry.

Conclusion

In this paper, we presented our redesign of a traditional one-on-one tutoring system to facilitate pair collaboration. The tutoring domain is Computer Science (CS) Education and the collaborative system takes advantage of the CS paradigm, pair programming. The paper presented our comparative analysis of two versions of the collaborative system which offer varying degrees of collaboration feedback to the pairs. We collected extensive logs of system use as well as audio recordings of pairs. We found that students in both conditions experienced significant learning gains. Moreover, shifts in dialogue initiative where significantly positively correlated to learning gains in both conditions. However, students provided with additional collaboration feedback in the *semistructured* condition, showed less signs of planning and symmetry as demonstrated through their time spent discussing the problem before coding and coding turn taking behavior.

References

Green, N., AlZoubi, O., Alizadeh, M., Di Eugenio, B., Fossati, D., & Harsley, R. (2015). A scalable intelligent tutoring system framework for computer science education. In *CSEDU 2015 - 7th International Conference on Computer Supported Education, Proceedings* (pp. 372–379). SciTePress.

Harsley, R. (2014). Towards a Collaborative Intelligent Tutoring System Classification Scheme. In *Proceedings Of The 11th International Conference On Cognition And Exploratory Learning In The Digital Age (Celda 2014)* (pp. 290–291). Porto, Portugal.

Harsley, R., Di Eugenio, B., Green, N., Fossati, D., & Acharya, S. (2016). Integrating Support for Collaboration in a Computer Science Intelligent Tutoring System. In *Proceedings of the 13th International Conference on Intelligent Tutoring Systems*. Croatia.

Janssen, J., Erkens, G., Kanselaar, G., & Jaspers, J. (2007). Visualization of Participation: Does It Contribute to Successful Computer-supported Collaborative Learning? *Comput. Educ.*, *49*(4), 1037–1065.

Kaptelinin, V. (1999). Learning together: Educational benefits and prospects for computer support. *Journal of the Learning Sciences*, *8*(3-4), 499–508.

Magnisalis, I., Demetriadis, S., & Karakostas, A. (2011). Adaptive and Intelligent Systems for Collaborative Learning Support: A Review of the Field. *IEEE Transactions on Learning Technologies*, *4*(1), 5–20.

Olsen, J. K., Aleven, V., & Rummel, N. (2015). Toward Combining Individual and Collaborative Learning Within an Intelligent Tutoring System. In C. Conati, N. Heffernan, A. Mitrovic, & M. F. Verdejo (Eds.), *Artificial Intelligence in Education* (Vol. 9112, pp. 848–851). Springer International Publishing.

Phielix, C., Prins, F. J., Kirschner, P. A., Erkens, G., & Jaspers, J. (2011). Group Awareness of Social and Cognitive Performance in a CSCL Environment: Effects of a Peer Feedback and Reflection Tool. *Comput. Hum. Behav.*, *27*(3), 1087–1102. https://doi.org/10.1016/j.chb.2010.06.024

Porter, L., Guzdial, M., McDowell, C., & Simon, B. (2013). Success in Introductory Programming: What Works? *Commun. ACM*, *56*(8), 34–36. https://doi.org/10.1145/2492007.2492020

Salleh, N., Mendes, E., & Grundy, J. (2011). Empirical Studies of Pair Programming for CS/SE Teaching in Higher Education: A Systematic Literature Review. *IEEE Transactions on Software Engineering*, *37*(4), 509–525. https://doi.org/10.1109/TSE.2010.59

Walker, M., & Whittaker, S. (1990). Mixed Initiative in Dialogue: An Investigation into Discourse Segmentation. In *Proceedings of the 28th Annual Meeting on Association for Computational Linguistics* (pp. 70–78). Stroudsburg, PA, USA: Association for Computational Linguistics.

Washington, A. N., Burge, L., Mejias, M., Jean-Pierre, K., & Knox, Q. (2015). Improving Undergraduate Student Performance in Computer Science at Historically Black Colleges and Universities (HBCUs) Through Industry Partnerships. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (pp. 203–206). New York, NY, USA: ACM.