Integrating Support for Collaboration in a Computer Science Intelligent Tutoring System

Rachel Harsley[1(✉)], Barbara Di Eugenio[1], Nick Green[1], Davide Fossati[2], and Sabita Acharya[1]

[1] Department of Computer Science, University of Illinois at Chicago, Chicago, IL, USA
{rharsl2,bdieugen,ngreen21,sachar4}@uic.edu

[2] Department of Math and Computer Science, Emory University, Atlanta, GA, USA
davide.fossati@emory.edu

Abstract. Calls for widespread Computer Science (CS) education have been issued from the White House down and have been met with increased enrollment in CS undergraduate programs. Yet, these programs often suffer from high attrition rates. One successful approach to addressing the problem of low retention has been a focus on group work and collaboration. This paper details the design of a collaborative ITS (CIT) for foundational CS concepts including basic data structures and algorithms. We investigate the benefit of collaboration to student learning while using the CIT. We compare learning gains of our prior work in a non-collaborative system versus two methods of supporting collaboration in the collaborative-ITS. In our study of 60 students, we found significant learning gains for students using both versions. We also discovered notable differences related to student perception of tutor helpfulness which we will investigate in subsequent work.

Keywords: Collaboration · Collaborative intelligent tutoring system · Data structures · CS1 · CS2

1 Introduction

In recent years, ITS researchers have begun to explore outcomes of ITSs that support collaborative learning. Benefits of collaborative learning include increased group performance as well as individual performance. Moreover, collaborative problem solving is consistently associated with higher order thinking skills including planning, reflection, and metacognition [5]. The field of Computer Supported Collaborative Learning (CSCL) explores how students learn in collaborative settings and how technology can support this collaboration.

There are a plethora of methods for system design regarding pedagogical guidance, group formation, collaboration cues, and student modeling in order for ITSs to accommodate collaboration [3]. Thus, we distinguish collaboration

A. Micarelli et al. (Eds.): ITS 2016, LNCS 9684, pp. 227–233, 2016.
DOI: 10.1007/978-3-319-39583-8_22

supported by a CIT in three primary ways: *unstructured* (initiated and maintained by students), *semistructured* or *fully structured* (moderately or strongly supported and guided by the CIT). This paper explores the role of the ITS in structuring collaboration by presenting findings from an empirical study in which students use the *unstructured* and *semi-structured* collaborative adaptations of a traditional ITS. We assess the effectiveness of the systems in terms of student learning gain and perceptions of the system. Findings are presented from a study with 60 students utilizing Collab-ChiQat Tutor, a collaborative ITS for computer science education. Results show that students using the *unstructured* system with minimal collaboration support, and the *semistructured* which provided collaboration feedback, both achieved significant learning gains.

2 Background

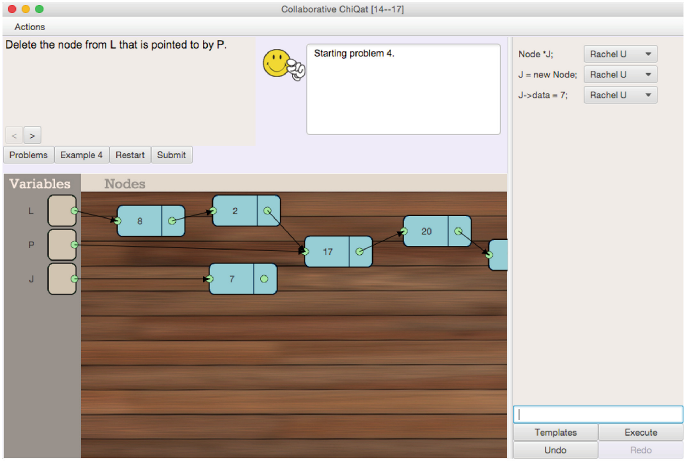
Longstanding research has shown that both cooperative and collaborative interactions among students are beneficial to learning [6]. However, assigning students to a group and charging them with a task does not ensure that students will engage in effective collaborative learning behavior [9]. Thus, CSCL requires careful construction of the collaboration so that interactions benefit the individual and group. One successful approach to improving collaboration has been the use of visualized group performance and peer assessments [4,8].

Collaboration is also a core component of CS curriculum and accreditation requirements [1]. It is been utilized in both industry and academia through the growing practice of pair programming. In this methodology, two users share the same computer, keyboard and mouse. One user serves as the driver while the other serves as the navigator. The driver's roles is to write the code and control both keyboard and mouse. The navigator's role is to act as an external metacognizer who thinks about the direction of the code and helps the pair avoid possible pitfalls.

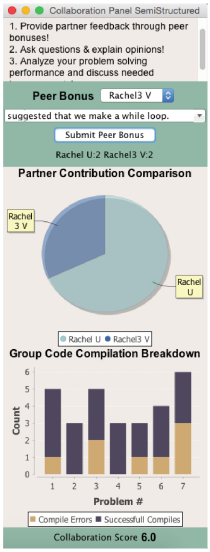
Recently, research efforts have focused on merging the affordances of both ITS and CSCL to capitalize on the benefits of group learning and adaptive support. Several researchers in the CSCL community are exploring how adaptivity, automated analysis, and feedback integrate into CSCL approaches [10]. Similarly, ITS researchers are extending their individual use ITS systems to accommodate collaborative support [7,11].

3 Collab-ChiQat Tutor

This study both reconceptualizes and redevelops a non-collaborative tutoring system for CS Education, ChiQat-Tutor. In particular our work centers on the system's linked lists data structure lesson. A problem is presented to a student in both textual and graphical representation as shown in Fig. 1a. The student is then able to programmatically solve the problem. Moreover, the system provides relevant positive and negative feedback to the student in a manner analogous to the one-on-one human tutoring experience from which the system was derived.



(a) Colab-ChiQat System



(b) Collaboration Panel

Fig. 1. Collab-ChiQat ITS for computer science education

Collab-ChiQat accommodates learning between pairs of students as they jointly engage with the system in pair programming. Collab-ChiQat maintains all of the major architectural components present in standard ChiQat. However, the collaborative system differs from the standard version in several ways including its student model, graphical user interface, and feedback.

In the *unstructured* version, students focus on CS domain learning with no system-provided support for their collaborative interaction. While in the *semi-structured* version, students focus on CS domain learning and have visualized representation of their participation and performance via the collaboration panel described below. Several newly introduced components for Collab-ChiQat are described below while our prior work sets forth existing components [2].

Joint Student Model. The joint student model works as the storehouse of information pertaining to a student's problem solving behavior and the state of the pairs' problem solving. The collection of information available in both the joint and individual student models is used to synthesize relevant and properly timed feedback. Information aggregated in the joint student module includes: history and timing of students' actions, feedback (*i.e.* number of positive/negative proactive/reactive feedback), undo/redo behavior, number of problem attempts and problems solved, individual and collective compile error and success rates, number of spoken utterances, peer bonus information.

Graphical User Interface. In *semistructured* Collab-ChiQat, a collaboration panel is introduced. The panel serves as the view for participation and group performance visualization and peer feedback as shown in Fig. 1b. The panel contains the following five components (1) tips on successful collaboration (2) pie chart comparison of number of spoken utterances between partners (3) bar graph comparison of number of compile errors vs successes per problem (4) peer bonus input w/sentence opener (5) overall group collaboration score.

4 Empirical Study

An experiment involving human participants was conducted in Fall of 2015 in a second year Computer Science programming course. Our experiments ran over four different sessions of the course. A total of 103 students used Collab-ChiQat during the study.[1] Students chose their own partners. Each pair was stationed at a single workstation and individually equipped with a headset. They were given 40 min to work with the system. Student interaction with the system was continually logged. Students were given an exit survey regarding their perception of Collab-ChiQat and their abilities, their attitudes towards CS and the course, and their understanding of successful pair programming traits.

Students were allowed 12 min to perform pre and post tests individually. Both pre and post tests are identical and derived from prior work analyzing human CS tutoring dialogues. We use the following measure of learning gain to assist in our analysis of learning:

$$gain = postTestScore - preTestScore \quad (1)$$

5 Results

Of foremost importance in evaluating the system is the answer to the question of whether or not students learned. In answer to the primary question, the students did learn. Overall, student post test scores were significantly better than pre-test scores ($p<.05$). Moreover, the learning gain in the *unstructured* condition approaches both our best prior results for the single student ChiQat system as well as the human tutoring[2] condition as shown in Table 1. Note, this holds true despite students' higher prior knowledge, given pre-test scores.

Subsequent to learning gains, our aim was to understand student perceptions of the system as captured through the exit survey. We were especially interested in student perception of system helpfulness. Contrary to our hypothesis, we discovered that a greater majority of students in the *unstructured* system condition found the system to be helpful than in the *semistructured* condition.

[1] 43 students had used the non-collaborative ChiQat in a prior experiment. Their data is held out from learning gain analysis and reserved for further work.

[2] The human tutoring condition measured learning gains of students after one 40-minute session of working with an experienced human computer science tutor.

Table 1. Learning gains of students

Tutor	N	Pre-test		Post-test		Gain	
		μ	σ	μ	σ	μ	σ
Human	54	.40	.26	.54	.26	.14	.25
Non-collaborative ChiQat (Best)	23	.41	.18	.55	.22	.14	.17
Unstructured Collab-ChiQat	30	.48	.21	.60	.22	.12	.17
Semistructured Collab-ChiQat	30	.52	.26	.61	.24	.08	.18

Further, student were asked to describe three attributes of a good pair programming partnership. Phrases such as "hard work" and "hard" appeared multiple times in the *semistructured* condition student feedback but did not appear at all in the *unstructured* condition feedback.

6 Discussion

The findings indicate that collaborative learning in conjunction with an ITS can enhance student learning. Results showed significant learning for students using both the *unstructured* collaborative system and the *semistructured* condition, which provided collaboration feedback. The findings are a crucial step toward applying known CSCL techniques, including visualized participation and peer feedback, to an ITS. Analysis of student feedback showed that students found the *semistructured* system less helpful and harder to use. There are several possible reasons for this student perception. First, the *semistructured* interface, which visualized individual participation and group performance, may have caused students to experience cognitive overload. Secondly, students may have also been disincentivized to perform well if under the impression that they were given "hard work" by the addition of the collaboration panel.

Future work will incorporate students removed from this study due to their prior exposure to non-collaborative ChiQat. Investigation of their results may shed light on the student's cognitive overload due to their increased familiarity with the overall system. Fine-grained analysis of interaction data including transcribed student interactions will also provide further insight regarding student perceptions of the system.

7 Conclusion

Collaborative Intelligent Tutoring Systems (CITs) offer a promising method to enhance student learning in adaptive and connected ways. In this paper, we detailed the design of an enriched architecture, a CIT for CS Education. In order to gain an understanding of the varying methods for supporting collaboration and their effect on learning, we compared two methods of structuring collaboration in a second year undergraduate CS course and analyzed student learning gains and system perceptions. We discovered that students found the *unstructured* version of the system, which provided no visualization of collaborative and individual performance, to be more helpful. They also experienced significant learning gains. Similarly, students in the *semistructured* condition experienced significant learning gain, however they found the system to be less helpful despite the additional participation and performance visualization.

Additional research is needed to understand how modes of supporting collaboration affect learning and social participation. Our future work will examine reasons for the learning gain disparity, including the possibility of introduced cognitive overload given the visualized feedback. It will become increasingly important to understand how CITs can provide support for students to effectively collaborate and learn.

Acknowledgments. This work was supported by the Abraham Lincoln Fellowship 2015–2016 from the University of Illinois at Chicago, and grant NPRP 5–939–1–155 from the Qatar National Research Fund.

References

1. Commission, A.C.A.: Criteria for Accrediting Computing Programs, 2016–2017 — ABET. ABET, November 2014
2. Green, N., AlZoubi, O., Alizadeh, M., Di Eugenio, B., Fossati, D., Harsley, R.: A scalable intelligent tutoring system framework for computer science education. In: Proceedings of the 7th International Conference on Computer Supported Education (CSEDU 2015) (2015)
3. Harsley, R.: 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, October 2014
4. Janssen, J., Erkens, G., Kanselaar, G., Jaspers, J.: Visualization of participation: does it contribute to successful computer-supported collaborative learning? Comput. Educ. **49**(4), 1037–1065 (2007)
5. Kaptelinin, V.: Learning together: educational benefits and prospects for computer support. J. Learn. Sci. **8**(3–4), 499–508 (1999)
6. Lehtinen, E., Hakkarainen, K., Lipponen, L., Rahikainen, M., Muukkonen, H.: Computer supported collaborative learning: a review. JHGI Giesbers Rep. Educ. **10** (1999)
7. Olsen, J.K., Belenky, D.M., Aleven, V., Rummel, N.: Using an intelligent tutoring system to support collaborative as well as individual learning. In: Trausan-Matu, S., Boyer, K.E., Crosby, M., Panourgia, K. (eds.) ITS 2014. LNCS, vol. 8474, pp. 134–143. Springer, Heidelberg (2014)
8. Phielix, C., Prins, F.J., Kirschner, P.A., Erkens, G., Jaspers, J.: 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 (2011)
9. Soller, A.: Supporting social interaction in an intelligent collaborative learning system. Int. J. Artif. Intell. Educ. (IJAIED) **12**, 40–62 (2001)

10. Tchounikine, P., Rummel, N., McLaren, B.M.: Computer supported collaborative learning and intelligent tutoring systems. In: Nkambou, R., Bourdeau, J., Mizoguchi, R. (eds.) Advances in Intelligent Tutoring Systems. SCI, vol. 308, pp. 447–463. Springer, Heidelberg (2010)
11. Walker, E., Rummel, N., Koedinger, K.R.: Integrating collaboration and intelligent tutoring data in the evaluation of a reciprocal peer tutoring environment. Res. Pract. Technol. Enhanced Learn. **04**(03), 221–251 (2009)