Collab-ChiQat: A Collaborative Remaking of a Computer Science Intelligent Tutoring System

Rachel Harsley
University of Illinois at Chicago
Chicago, IL 60607, USA
rharsl2@uic.edu

Satabdi Aditya
University of Illinois at Chicago
Chicago, IL 60607, USA
sachar4@uic.edu

Nick Green
University of Illinois at Chicago
Chicago, IL 60607, USA
ngreen21@uic.edu

Davide Fossati
Carnegie Mellon University Qatar
Doha, Qatar
dfossati@cmu.edu

Barbara Di Eugenio
University of Illinois at Chicago
Chicago, IL 60607, USA
bdieugen@uic.edu

Omar AlZoubi
Carnegie Mellon University Qatar
Doha, Qatar
oalzoubi@cmu.edu

CSCW '16 Companion, February 27 - March 02, 2016, San Francisco, CA, USA
ACM 978-1-4503-3950-6/16/02.
http://dx.doi.org/10.1145/2818052.2869118

Abstract
This paper focuses on the motivation, design, and initial prototype implementation of Collab-ChiQat. Collab-ChiQat is a collaborative reconceptualization of an existing intelligent tutoring system for Computer Science Education originally intended for one-to-one student-system tutoring. Collab-ChiQat allows students to work as pair programmers as they solve coding problems for linked lists, a foundational and difficult to grasp CS concept. The work is unique in it's comparison of how system structuring of collaboration affects both learning and actual collaboration. In one condition, students are left to themselves with no system feedback regarding their collaborative behavior. While in a second condition, the collaboration is semi-structured, meaning students received a visualization of their participation and other metrics.

Author Keywords
Intelligent Tutoring Systems; Collaboration; Computer Science Education; Pair Programming; Linked Lists

ACM Classification Keywords
H.5.3 [Information Interfaces and Presentation]: Group and Organization Interfaces – Computer-supported cooperative work.

Background

The demand for workers with CS skills is at an all-time high [7]. However, many students struggle to grasp foundation CS concepts and data structures including linked lists and their manipulation. In industry, the practice of pair programming has seen an increase. In pair programming, two programmers work side by side at one computer to collaboratively design and write code. While, ITSs provide adaptive, user-centric feedback to students as they learn, standard definitions of an ITS often limit the scope of the tutor to interactions and modeling of an individual learner [1]. On the other hand, CSCL research has shown that, assigning students to a group and charging them with a task does not ensure that students will engage in effective collaborative learning behavior [4,8]. CSCL applications structure group activity in order to promote these effective collaborative behaviors.

Introduction

A primary reason for the effectiveness of human tutoring rests in the tutor's ability to accurately assess a student's knowledge on an ongoing basis, adaptively share expertise, and do this all in real-time. In the same way, intelligent tutoring systems (ITS) provide real-time feedback and adaptively adjust to student knowledge [9]. However, the ITS has traditionally been geared toward one-to-one student learning, despite the noted benefits of group, or collaborative, learning [5]. Originally a constraint due to physical system ability, this feature of the one-to-one model is baked into the framework of ITS literature [7]. The aim of our research is to shift this paradigm for an existent ITS for CS Education in order to accommodate collaborative learning via pair programming.

Collab-ChiQat

Collab-ChiQat reconceptualizes the architecture and implementation of the standard, non-collaborative, computer science intelligent tutoring system, ChiQat-Tutor. Collab-ChiQat deals exclusively with the linked list learning module of the standard system. Furthermore, Collab-ChiQat accommodates learning between pairs of students as they jointly engage with the system. Collab-ChiQat maintains all of the major architectural components present in standard ChiQat. However, the collaborative system differs from the standard version in its design of the graphical user interface and student model. This is due to Collab-ChiQat's added objective which is to measure, evaluate, and provide feedback regarding collaborative behavior among peers in addition to facilitating domain learning.

Two Collaboration Structures

Given the range of approaches to supporting collaboration in an ITS [3], one of Collab-ChiQat's main research goals is to explore how different types of collaboration structures affect students' CS learning as well as other measures of collaboration. Thus, two collaboration structuring types were implement as follows:

UNSTRUCTURED: situates students as pair programmers working with the tutoring system. Furthermore, with each line of code submitted, students must specify who was acting as the driver (owner). The system provides no feedback to students on the collaboration itself.

SEMI-STRUCTURED: provides students with passive feedback regarding their collaboration. Students also specify code ownership. The feedback includes:

- Helpful tips on successful collaboration ie: *"Ask questions and explain opinions"*
- Pie chart comparison of the number of spoken utterances, code ownership, and peer bonuses per partner
- Code compilation error vs success rate per problem
- Peer Bonus sentence opener ie: "_____ *encouraged me by* _____"
- Collaboration Score

System Architecture

Collab-ChiQat's primary purpose is to ameliorate a student's learning of the linked list data structure [2]. A problem is presented to a student in both textual and graphical representation. The student is

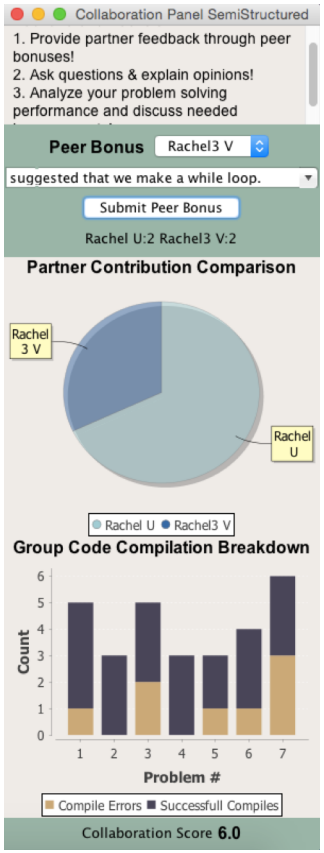

Figure 3: Semi-structured Collab-ChiQat introduces a collaboration panel. The panel serves as the view for collaboration visualization, input, feedback and instructions. The collaboration panel contains both graphs to represent collaboration metrics and textual-based output.

then able to programmatically solve the problem. Moreover, the system provides relevant feedback to the student in a manner analogous to the one-on-one human tutoring experience from which the system was derived. Example problem types involve linked list node insertion and removal in addition to other more complicated operations as shown in Figure 1.

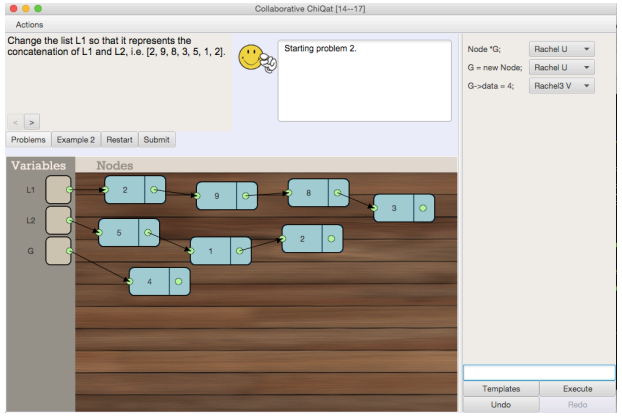

Figure 1: Collab-ChiQat's Main graphical interface which includes tutor feedback, problem panel, graphical state representation, and coding area including code owner.

Collab-ChiQat comprises six major components consistent with the standard, non-collaborative version of ChiQat as follows: graphical user interface, problem model, constraint evaluator, feedback manager, procedural knowledge model, and student model. Due to Collab-ChiQat's added objective of collaboration structuring, a collaboration feedback manager and collaborative interaction model are also introduced to the system architecture. The architecture of Collab-ChiQat is depicted in Figure 2.

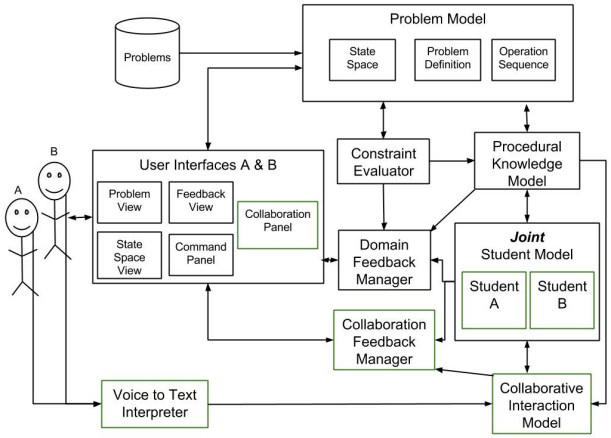

Figure 2: Collab-ChiQat system architecture with additions from standard ChiQat including; Voice to Text, Collaboration Feedback Manager, Collaborative Interaction Model, Joint Student Model and Collaboration Panel.

Initial Feedback

108 students from a second year CS programming practicum course used Collab-ChiQat for 40 minutes during a single lab session as seen in Figure 4. Student feedback revealed the vast majority of students found working with Collab-ChiQat to be interesting and helpful as seen in Figure 5. This majority was larger than prior studies dealing with single user, standard ChiQat (59% and 53% of users agreed or strongly agreed). There was also a notable difference between the perceived helpfulness reporting for the unstructured and semi-structured conditions. Surprisingly, a greater majority of students in the unstructured condition found the system to be more helpful than in the semi-structured condition as seen in Table 1.

Figure 4: Students using Unstructured Collab-ChiQat which did not include the Collaboration Panel.

	Unstruc.	Semi.
strongly disagree	2	3
disagree	6	3
neutral	8	13
agree	25	19
strongly agree	13	12
TOTAL	54	50

Table 1: A comparison of survey results for prompt "Do you feel that ChiQat-Tutor helped you learn about linked lists?".

Figure 6: Students using Semi-Structured Collab-ChiQat which includes the Collaboration Panel.

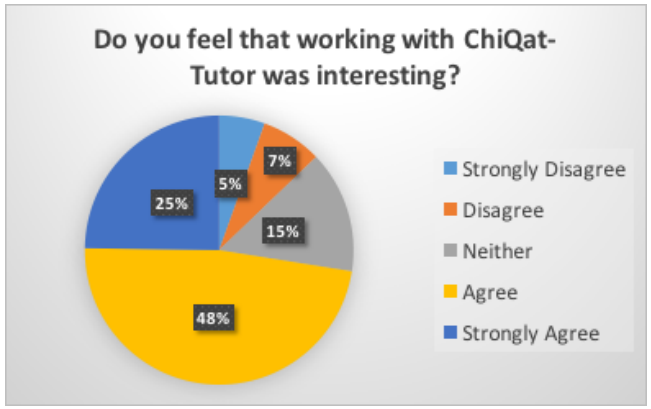

Figure 5: Student feedback from exit survey.

Conclusions & Future Work

In this informative poster, we have outlined the motivation for our collaborative ITS, Collab-ChiQat. Collab-ChiQat is a unique ITS because it deviates from the traditional mode of ITS support limited to a single student. The system was used across two modes of collaboration structuring, one with no system feedback on collaboration, the other with collaboration visualization. Students reported more interest in this system than our previous single-user interventions, however, they found the unstructured version to be more helpful. We will move forward in examining how domain learning was affected across the two conditions as well as other outcome measures.

Acknowledgments

This publication was made possible by NPRP grant 5-939-1-155 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

1. Cristina Conati. 2009. Intelligent Tutoring Systems: New Challenges and Directions. Proceedings of the 21st International Jont Conference on Artifical Intelligence, Morgan Kaufmann Publishers Inc., 2–7.
2. Davide Fossati. 2013. ChiQat: An intelligent tutoring system for learning computer science. Qatar Foundation Annual Research Forum Proceedings, 2013: ICTP 020.
3. Rachel Harsley. 2014. Towards a Collaborative Intelligent Tutoring System Classification Scheme,. Proceedings Of The 11th International Conference On Cognition And Exploratory Learning In The Digital Age (Celda 2014), 290–291.
4. Lars Kobbe, Armin Weinberger, Pierre Dillenbourg, et al. 2007. Specifying computer-supported collaboration scripts. International Journal of Computer-Supported Collaborative Learning 2, 2-3: 211–224.
5. Erno Lehtinen, Kai Hakkarainen, Lasse Lipponen, Marjaana Rahikainen, and Hanni Muukkonen. 1999. Computer supported collaborative learning: A review. The JHGI Giesbers reports on education 10.
6. Matt Richtel. 2014. Reading, Writing, Arithmetic, and Lately, Coding. The New York Times.
7. J.W. Rickel. 1989. Intelligent computer-aided instruction: a survey organized around system components. IEEE Transactions on Systems, Man and Cybernetics 19, 1: 40–57.
8. Amy Soller. 2004. Computational Modeling and Analysis of Knowledge Sharing in Collaborative Distance Learning. User Modeling and User-Adapted Interaction 14, 4: 351–381.
9. Pierre Tchounikine, Nikol Rummel, and Bruce M. McLaren. 2010. Computer Supported Collaborative Learning and Intelligent Tutoring Systems. In Advances in Intelligent Tutoring Systems. Springer, 447–463.