12th International Conference on Intelligent Tutoring Systems
Honolulu, Hawaii, June 5-9, 2014

Authoring Tools for Collaborative Intelligent Tutoring System Environments

Jennifer K. Olsen[1], Daniel M. Belenky[1], Vincent Aleven[1], Nikol Rummel[12], Jonathan Sewall[1], and Michael Ringenberg[1],

[1]Human Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA, USA
jkolsen@cs.cmu.edu, dbelenky@andrew.cmu.edu,
{aleven,sewall,mringenb}@cs.cmu.edu
[2]Institute of Educational Research, Ruhr-Universität Bochum, Germany
nikol.rummel@rub.de

Abstract. Authoring tools have been shown to decrease the amount of time and resources needed for the development of Intelligent Tutoring Systems (ITSs). Although collaborative learning has been shown to be beneficial to learning, most of the current authoring tools do not support the development of collaborative ITSs. In this paper, we discuss an extension to the Cognitive Tutor Authoring Tools to allow for development of collaborative ITSs through multiple synchronized tutor engines. Using this tool, an author can combine collaboration with the type of problem solving support typically offered by an ITS. Different phases of collaboration scripts can be tied to particular problem states in a flexible, problem-specific way. We illustrate the tool's capabilities by presenting examples of collaborative tutors used in recent studies that showed learning gains. The work is a step forward in blending computer-supported collaborative learning and ITS technologies in an effort to combine their strengths.

Keywords: Problem solving, collaborative learning, intelligent tutoring system, authoring tools

1 Introduction

While most Intelligent Tutoring Systems (ITSs) are geared towards individuals, there has been some evidence that collaborative ITSs are also beneficial [5-6], [14]. ITSs take advantage of features, such as step-based guidance and hints, to support successful learning [12] while Computer Supported Collaborative Learning (CSCL) environments provide support for learning through collaboration scripts, which provide structure for tasks and interactions within a group, and help support the development of mutual understanding and explanation [3]. Despite these benefits, the combination of the two may not be more widely used because of a lack of effective and flexible authoring tools for creating collaborative learning opportunities within ITSs [8]. While there has been ongoing work to develop collaboration tools to make collaboration scripts more accessible and easier to use across learning domains [1], [7], [11], [13], these tools often do not take advantage of beneficial ITS features. We have created a tool that flexibly supports the use of collaboration scripts while also providing support for ITS features by extending an existing ITS authoring tool, the Cognitive Tutor Authoring Tools (CTAT) [2].

To demonstrate the utility of the tool, we will present examples from experiments we have run where we have created learning opportunities based on *collaboration scripts*. According to Dillenbourg [3], a collaboration script consists of a set of phases where each phase has five attributes: the task, the group composition, the distribution of the task (this includes who gets what information *and* who does what, such as through roles), the mode of the interaction, and the order of the phases. Any of these attributes can change between phases, and to allow for flexibility in the scripts developed, an authoring tool needs to support each of these attributes independently so the script can dynamically change with the problem state.

The enhancement to CTAT described in this paper supports the development of ITSs that contain these attributes. Authors can create collaborative ITSs by embedding various problem-specific features that trigger dynamically, based on the problem state, to move students through different phases of the collaboration script, all without programming. In this paper, we provide examples of collaborative script phases (i.e. cognitive group awareness [4] and sharing unique information) developed using CTAT. These examples were used in two "pull-out studies," run in three elementary schools, with a total of about 70 participating students in collaborative conditions and illustrate the flexibility of authoring collaborative tutors.

2 Authoring Tool Extensions to Support Collaboration

In this section, we describe how one type of ITS, a collaborative example-tracing tutor [2], can be authored with CTAT. Similar to how tutors for individual learners are developed, an author creates two key components: A user interface designed for the problem being tutored (in Flash) and a behavior graph (in the CTAT software), which stores all of the acceptable solution paths and commonly-occurring incorrect steps. Behaviorally, example-tracing tutors are similar to other types of ITSs, providing all the key functionality defined by VanLehn [12], and below we describe how the CTAT extension has allowed communication between tutors for collaboration.

2.1 Authoring Collaborative Tutors

To expand CTAT so it supports *collaborative example-tracing tutors*, we added the capability to run *multiple synchronized tutor engines,* one for each student in a collaborating group (see Figure 2). It is important to note that any number of tutor engines can be run in synchronized fashion. Specifically, for any given problem in a collaborative tutor, there is a separate behavior graph file per collaborating student and a separate interface file. The collaborative version of CTAT allows authors to synchronize the tutors so that it can maintain a problem state that is in sync between tutor engines (and between collaborating students). When one of the collaborating students takes an action, this input is sent to both the student's tutor engine and their partner's tutor engine. By contrast, tutor output is only sent to the corresponding student interface. One result of this input sharing is that student actions taken on one interface will be "mirrored" on the other interface in the corresponding interface component. Yet this set up also allows for differentiation in the tutor feedback provided to collaborating partners, for example by means of unique feedback, individualized hints, information based on roles, and different sets of available actions at any given point in a problem. This set-up allows for great flexibility in authoring tutors with embedded collaboration scripts. In particular, the power of the approach comes from being able to craft tutors in which the collaborators have different views on the same problem and tasks are distributed across collaborators, so as to structure and support their different roles according to particular collaborative phases in a collaboration script. There are many collaboration features, such as the cognitive group awareness and unique information described below, as well as other scripts such as the jigsaw and the tutee/tutor paradigm, where the benefit of the activity comes from the students having different roles and responsibilities in the problem-solving task.

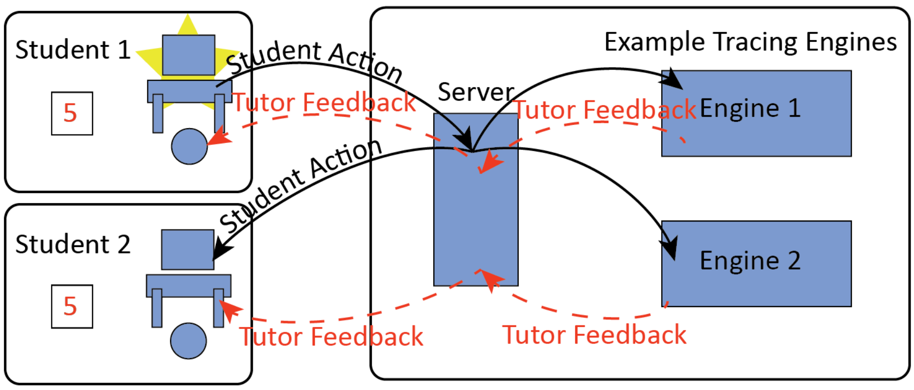

Fig. 2. Diagram displaying the communication between two synchronous tutor engines. A student interface action is shown with the solid line and feedback is shown with the dotted line. Student 1 has entered a 5 into the interface, which has been distributed to both example-tracing engines, and each student has received individual feedback based on the result.

To author a collaborative tutor, each of the steps to create an individual tutor is followed for each member of the collaboration, typically, in interleaved fashion. First, an author creates an interface through drag-and-drop with Flash. Each interface can be identical or designed to match the student's roles. Once an interface is created, an author creates a behavior graph by demonstrating problem-solving behavior on the interface. After the behavior graph is created through demonstration, the graph can be annotated with hints and error feedback messages. The hints and feedback provided can be the same for each student or can be customized for each student. To author collaborative tutors, CTAT allows multiple behavior graphs to be open simultaneously and to each connect to their own student interface. This allows authors to test the collaboration and synchronization of the tutor engines.

3 Collaboration Examples Using CTAT for Collaboration

Below we describe two examples of collaborative phases, which have been shown to be successful [4], to demonstrate the flexibility of CTAT in supporting different collaboration features. Specifically, building on our prior work on the Fractions Tutor [10], we created a collaborative tutoring system to help elementary students learn fractions. In three school studies, we have observed positive learning gains related to tutor usage [9]. As students use the tutor, they each sit at their own computer and communicate via Skype. The two examples illustrate the types of collaboration features that can be implemented within a ITS using the collaborative version of CTAT.

The first example demonstrates a task that supports cognitive group awareness, in which the students are learning conceptual knowledge about equivalent fractions. Cognitive group awareness refers to having information about group members' knowledge, information, or opinions, and sharing of this information has been shown to help guide collaboration [4]. In this example, cognitive group awareness is combine with step-by-step support for problem solving as follows: First, the collaborating partners each answer the same question separately; then, the tutor displays both partners' answers to promote discussion; and, finally, the partners provide a final answer endorsed by both (see Figure 1, panel A1). The students are not given feedback on their individual answer but are shown what their partner selected and are asked to select the correct answer as a pair. This allows each student to see their partner's understanding of the question before discussing and choosing a group answer.

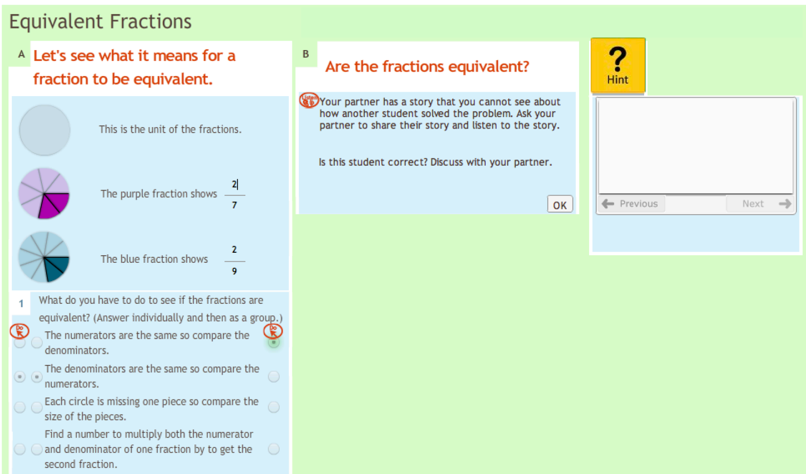

Fig. 1. Example conceptual tutor problem. Panel A1 displays an example of support for cognitive group awareness. Panel B displays an example of individual information.

We also used the enhanced version of CTAT to implement a second collaboration script phase, in which students are provided with unique information to share with their partner. As in the previous example, the collaborative tutor provides a different view on the same problem for each collaborating partner. Specifically, we implemented a script that distributes information between the partners and supports the sharing of this information. Students are either shown an example response about the fractions and asked to share with their partner, as indicated by the "share" icon, or are asked to listen to their partner's information, as indicated by the "listen" icon (see Figure 1, panel B). After the first student shares their example response, the students then switch roles, with the second student receiving a different example to share. This activity provides each student with a different viewpoint that they can then use to start a discussion. Both example phases illustrate a range of collaborative tasks that can be supported using CTAT for collaboration by integrating the group formations (individual or dyadic tasks), the task distributions (roles and unique information), and the timing of the phase for the different tasks (ordering of the tasks) into a ITS environment that can provide feedback and hints to the student.

4 Discussion and Conclusion

CSCL has been shown to be an effective paradigm for knowledge acquisition [6], yet most authoring tools for ITSs do not support collaborative learning. We extended CTAT so it supports the authoring of collaborative tutors, allowing for scripts to be *flexibly* developed to align with the problem state and goals, while maintaining the typical ITS advantages. With this new version of CTAT, authors can develop collaborative ITSs with embedded collaboration scripts, so that features that support effective collaboration can be intertwined with those that support problem solving and the support for collaboration and problem solving can unfold dynamically with the problem state and can be shared among collaborating students. Unlike many CSCL tools, the tutor follows along with the students and can provide personalized hints and feedback on domain knowledge.

The extension to support collaborative authoring required a relatively small number of changes to CTAT, although these changes enable a wide range of collaborative tutoring interactions to be authored. First, we made it possible to use multiple tutor engines in synchronized fashion. Each tutor engine "serves" a single student in a group, but has access to the actions of the other students. This loose coupling makes it possible for the tutor engines to maintain a shared problem state yet respond differently to each student. CTAT provides the flexibility to develop a wide range of scripts. Collaborative tutors built using the CTAT extension have been used successfully in two different studies [9].

ITS and CSCL work often proceed somewhat separately. The work reported here represents a step forward in blending certain ITS tools and CSCL tools, in an effort to combine their strengths. Authoring collaborative ITSs with CTAT works well for collaboration scripts closely tied to the problem state but does not support collaboration scripts that are more independent of the problem, such as conversational agents. Cognitive group awareness and unique information were given as examples in this paper, but the design space is much larger and limits are still being determined. We look forward to continued use of our combined tool in the ITS and CSCL communities to explore the range of collaborative tutoring interactions it can support.

Acknowledgments. We thank the Cognitive Tutor Authoring Tools team for their help. This work was supported by Graduate Training Grant # R305B090023 and by Award # R305A120734 both from the US Department of Education (IES).

5 References

1. Adamson, D., & Rosé, C. P.: Coordinating Multi-Dimensional Support in Collaborative Conversational Agents. In: Intelligent Tutoring Systems (pp. 346-351). Springer Berlin/Heidelberg (2012)
2. Aleven, V., McLaren, B. M., Sewall, J., & Koedinger, K. R.: A New Paradigm for Intelligent Tutoring Systems: Example-tracing Tutors. International Journal of Artificial Intelligence in Education, 19, 105-154 (2009)
3. Dillenbourg, P. Over-scripting CSCL: The risks of blending collaborative learning with instructional design. Three worlds of CSCL. Can we support CSCL?, 61-91 (2002)
4. Janssen, J., & Bodemer, D.: Coordinated Computer-Supported Collaborative Learning: Awareness and Awareness Tools. Educational Psychologist, 48(1), 40-55 (2013)
5. Lesgold, A., Katz, S., Greenberg, L., Hughes, E., & Eggan, G.: Extensions of intelligent tutoring paradigms to support collaborative learning. In S. Dijkstra, H. P. M. Krammer, & J. J. G. van Merrienboer (Eds.), Instructional models in computer-based learning environments. (pp. 291-311). Berlin: Springer-Verlag (1992)
6. Lou, Y., Abrami, P. C., & d'Apollonia, S.: Small group and individual learning with technology: A meta-analysis. Review of educational research, 71(3), 449-521 (2001)
7. Miao, Y., Hoeksema, K., Hoppe, H. U., & Harrer, A.: CSCL Scripts: Modelling Features and Potential Use. In Proceedings of the 2005 conference on Computer support for collaborative learning: learning 2005: the next 10 years! (pp. 423-432). ISLS (2005)
8. Murray, T., Blessing, S., & Ainsworth, S.: Authoring tools for advanced technology learning environments: Toward cost-effective adaptive, interactive and intelligent educational software. Amsterdam: Kluwer Academic Publishers (2003)
9. Olsen, J. K., Belenky, D. M., Aleven, A., & Rummel, N.: Using an intelligent tutoring system to support collaborative as well as individual learning. In: Intelligent Tutoring Systems. Springer Berlin/Heidelberg (in press)
10. Rau, M., Aleven, V., Rummel, N., & Rohrbach, S.: Sense Making Alone Doesn't Do It: Fluency Matters Too! ITS Support for Robust Learning with Multiple Representations. In: Intelligent Tutoring Systems, pp. 174-184. Springer Berlin/Heidelberg (2012)
11. Stegmann, K., Streng, S., Halbinger, M., Koch, J., Fischer, F., & Hußmann, H.: eXtremely Simple Scripting (XSS): A framework to speed up the development of computer-supported collaboration scripts. In Proceedings of the 9th International Conference on Computer supported Collaborative Learning-Volume 2 (pp. 195-197). ISLS (2009)
12. VanLehn, K.: The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. Educational Psychologist, 46, 197-221, (2011)
13. Wecker, C., Stegmann, K., Bernstein, F., Huber, M. J., Kalus, G., Rathmayer, S., Kollar, I., & Fischer, F.: Sustainable script and scaffold development for collaboration on varying web content: the S-COL technological approach. In Proceedings of the 9th int'l conference on Computer supported collaborative learning-Volume 1 (pp. 512-516). ISLS (2009)
14. Walker, E., Rummel, N., & Koedinger, K.: CTRL: A research framework for providing adaptive collaborative learning support. User Modeling and User-Adapted Interaction: The Journal of Personalization Research (UMUAI), 19(5), 387-431 (2009)