



The Effect of Metacognitive Scaffolding for Learning by Teaching a Teachable Agent

Noboru Matsuda¹  · Wenting Weng² · Natalie Wall³

Published online: 8 January 2020

© International Artificial Intelligence in Education Society 2020

Abstract

The effect of metacognitive scaffolding for learning by teaching was investigated and compared against learning by being tutored. Three versions of an online learning environment for learning algebra equations were created: (1) APLUS that allows students to interactively teach a synthetic peer with a goal to have the synthetic peer pass the quiz while the system provides students with metacognitive scaffolding on how to teach. (2) APLUS TUTOR that provides cognitive tutoring (i.e., immediate feedback and just-in-time hint) and metacognitive scaffolding on how to learn. And, (3) COGTUTOR+ that provides traditional cognitive tutoring on mastery learning. Two school studies were conducted with a total of 444 6th through 8th grade students. 208 students completed the study and were included in the analysis. The results show that (i) students' proficiency in solving equations increased after using our interventions for 4 days, but there was no difference in the effectiveness across three interventions, and (ii) learning by teaching with metacognitive scaffolding facilitated learning equally across various levels of students' prior competency.

Keywords Learning by teaching · Goal-oriented practice · Teachable agent · Algebra

Introduction

Learning by teaching is a promising style of learning that has been empirically studied with remarkable positive effects in many subject domains (Cohen 1994; Cohen et al.

✉ Noboru Matsuda
Noboru.Matsuda@ncsu.edu

¹ Department of Computer Science, North Carolina State University, 890 Oval Dr., Raleigh, NC 27606, USA

² Department of Educational Psychology, Texas A&M University, 540 Ross Street, College Station, TX 77843, USA

³ Department of Teaching, Learning and Culture, Texas A&M University, 540 Ross Street, College Station, TX 77843, USA

1982; Hedin 1987; Roscoe and Chi 2007). The literature shows that peer tutoring is an effective style of learning both for tutors and tutees—i.e., students learn by teaching others. The phenomenon is often called *tutor learning*. Various empirical studies have reported that tutor learning happens in various domains in various educational settings (Annis 1983; Devin-Sheehan et al. 1976; Gartner et al. 1971).

Learning by teaching is known to be an effective style of peer learning even with students at low proficiency levels (Britz et al. 1989; Robinson et al. 2005). Students can be trained to be better tutors, which amplifies the effect of tutor learning (Fuchs et al. 1997; King et al. 1998). Even simply switching tutor/tutee roles when receiving individual tutoring, aka reciprocal teaching, enhances learning (Palincsar and Brown 1984).

Despite its known effect, however, learning by teaching has some practical issues when it is applied to actual classroom instruction. For example, students must take turns to switch roles between tutor and tutee, which requires twice as much time as other types of instructions that allow students to do the same exercise simultaneously. Other researchers found that tutors learn at a cost of tutees, i.e., tutees might not learn as much as tutors do (King et al. 1998; Walker et al. 2014).

To develop a transformative educational technology to make learning by teaching practical in authentic learning settings, researchers have conducted studies using an online learning environment that allows students to learn by teaching a synthetic peer that is often called a *teachable agent* (Biswas et al. 2005; Schwartz et al. 2007). The effect of learning by teaching a teachable agent has been studied in various domains for different research questions as shown in the section for Related Works. However, the reported effects of learning by teaching a teachable agent have been mixed—some studies showed positive while others showed null effects.

We hypothesize that one reason for not seeing a stable effect with a teachable agent technology is a lack of an underlying cognitive theory as a design principle when building a teachable agent—i.e., a computational modeling perspective of learning by teaching that can be transformed into an online learning environment. Too little is known in the current literature about the critical factors for a successful implementation of a teachable agent to induce the expected effect of learning by teaching.

Though the literature review shown in the Related Works section is not exclusive, it provides some insight into the state-of-the-art knowledge about the effect of learning by teaching with teachable agent. Among many factors, based on the lessons learned from past studies, we are particularly interested in the effect of *adaptive scaffolding* while students are teaching their teachable agent. A study on Betty's Brain, for example, indicates the importance of student's self-regulation skills and adaptive guidance to scaffold students to learn those skills (Roscoe et al. 2013).

The current study is also motivated by and built on our previous studies to understand the effect of learning by teaching a teachable agent relative to learning by being tutored (aka cognitive tutoring) with a particular focus on the impact of *adaptive scaffolding*. We have developed an online learning environment, called APLUS—**A**rtificial **P**eer **L**earning environment **U**sing **S**im**S**tudent that allows students to learn to solve algebra linear equations (generally taught in 7th and 8th grade math) while teaching a teachable agent, called SimStudent (Matsuda et al. 2010). In our initial attempt (Matsuda et al. 2011), an earlier version of APLUS that did not have the adaptive scaffolding was compared with a commercial version of Carnegie Learning

Cognitive Tutor Algebra I™ that provides students with mastery learning on solving equations. While the 2011 study found no condition difference on the post-test score when the pre-test score was controlled, there was an aptitude-treatment interaction—learning by being tutored was more beneficial to students with low prior (measured as the pre-test score) whereas both treatments were equally beneficial to high prior students. An analysis of the interactions between students and the system showed that *students often taught the teachable agent incorrectly and inappropriately without knowing that they made such errors*. For example, students taught incorrect solutions, or taught only a particular type of problem when the teachable agent must learn different skills. Students with low prior were more prone to commit such suboptimal teaching behaviors than those with high prior.

To overcome this issue, we have developed a teacher agent (often called a meta-tutor) that provides students with adaptive scaffolding while they are teaching the synthetic peer and integrated it into the APLUS learning environment. One of the obvious research questions was about a kind of scaffolding the meta-tutor should provide for students. Based on our previous studies, we hypothesized that students need scaffolding on how to solve problems (cognitive scaffolding), how to teach (metacognitive scaffolding, which is described in detail in the “INTERVENTIONS” section), or both.

We first implemented the *metacognitive scaffolding* on the APLUS learning environment by embedding a meta-tutor agent that provides students with adaptive help on how to teach while they teach the teachable agent. Results from a classroom study showed that adding metacognitive scaffolding amplified the effect of learning by teaching more than the baseline learning environment where no metacognitive scaffolding was provided (Matsuda et al. 2014).

We then implemented *cognitive scaffolding* as a new functionality of the meta-tutor. The cognitive scaffolding was driven by the model-tracing technique commonly used by cognitive tutors (Anderson and Pelletier 1991). The meta-tutor with cognitive scaffolding provided students with adaptive help on how to solve equations. We conducted a further classroom study to compare the effect of cognitive and metacognitive scaffolding (Matsuda et al. 2016). *The results showed that metacognitive scaffolding (again) facilitates the effect of learning by teaching, but cognitive scaffolding does not*. This finding that cognitive scaffolding does not facilitate tutor learning is somewhat surprising given a well known effect of cognitive tutoring (Pane et al. 2014; Ritter et al. 2007). The literature also suggests that the effect of learning by teaching is rooted in the process of teaching preparation, aka the teaching expectancy principle (Renkl 1995), which conjectures the benefit of cognitive scaffolding.

The current study is to build on the knowledge on the effect of metacognitive scaffolding for tutor learning obtained from our three previous studies (Matsuda et al. 2016; Matsuda et al. 2014; Matsuda et al. 2011). The primary aim of the current paper is to understand how the presence of metacognitive scaffolding changes our knowledge about the relative effectiveness of learning by teaching against cognitive tutoring. In the following text, we use the words “cognitive tutoring” and “learning by being tutored” interchangeably.

Although, it is important to understand relative effectiveness among different learning strategies to advance the theory of how people learn, learning by teaching has been rarely compared with other types of learning strategies in the current literature.

Learning by teaching has been mostly compared with studying by reading a text (Annis 1983) or regular classroom instructions (Sharpley et al. 1983; Zhao, and Ailiya, and Shen, Z. 2012). Klingner and Vaughn (1996) is one of the rare examples where learning by teaching is compared with reciprocal teaching. Empirical studies with teachable agents often only control functionality of the teachable agent hence comparing learning by teaching with one characteristic against another characteristic—e.g., the amount of prior knowledge (Jun 2003), the presence of adaptive scaffolding by the meta-tutor (Matsuda et al. 2016; Matsuda et al. 2014; Roscoe et al. 2013; Tan et al. 2006), the availability of a competitive gameshow (Matsuda et al. 2013a), and motivational incentive (Uresti and du Boulay 2004). The current work will therefore contribute to the literature by yielding a new insight into the effect of learning by teaching relative to cognitive tutoring.

For the current study, we are particularly interested in answering the following specific research question—*does the aptitude-treatment interaction among learning by teaching and learning by being tutored that we observed in the previous study still exist with the presence of metacognitive scaffolding?* To answer this question, the current paper reports two classroom studies conducted at three public schools in their business-as-usual settings. The effect of three types of learning strategies—i.e., learning by teaching (APLUS with metacognitive scaffolding) was compared with the direct instruction (two versions of cognitive tutors with and without metacognitive scaffolding). The direct instruction was driven by a cognitive tutor (Anderson et al. 1985), which provides a mastery learning on prespecified set of skills.

The effect of cognitive tutoring has been intensively studied (Anderson et al. 1995; Pane et al. 2014; Ritter et al. 2007). However, very little is known about the comparison between learning by being tutored and learning by teaching. As far as we are aware of, Biswas et al. (2010) and Matsuda et al. (2011) are the only controlled studies in authentic learning environment where learning by teaching a teachable agent was compared with direct instructions (see the Related Works section for more details).

To investigate the effect of metacognitive scaffolding, we need a tight control between learning by teaching and the direct instruction. When it comes to the implementation level, APLUS and a cognitive tutor have some notable difference other than the availability of metacognitive scaffolding. The next section provides an overview of how we operationalize learning by teaching and learning by being tutored.

A Tight Comparison between Three Proposed Interventions

The version of APLUS used in the current study is an extension of the one used in our previous study (Matsuda et al. 2011) with the meta-tutor added to provide metacognitive scaffolding. To make a tightly controlled comparison, we developed a version of cognitive tutor, COGTUTOR+, that looks essentially identical to APLUS (section “INTERVENTIONS” provides details about our interventions). We designed COGTUTOR+ in such a way that it closely mimics the tutoring behavior of traditional cognitive tutors (VanLehn 2006)—i.e., the adaptive *cognitive* scaffolding (i.e., immediate feedback and just-in-time hint) driven by model tracing (Anderson et al. 1990) and the adaptive problem selection based on knowledge tracing for mastery learning (Corbett and Anderson 1995).

However, using COGTUTOR+ as a comparative intervention for APLUS yields some concerns for potential confounding, at least for the following two factors: (1) an agent who selects problems for practice—a tutor vs. a student, and (2) a criterion for graduation from practice—a student reaching to a mastery level vs. a teachable agent passing a quiz. For the problem selection, traditional cognitive tutors select problems adaptively based on a student’s competency. On the other hand, when using APLUS, students select problems to teach their synthetic peers, apparently based on the synthetic peer’s perceived competency. For the graduation criterion, traditional cognitive tutors estimate a student’s competency in applying skills and stop the practice when the estimation exceeds a given threshold. On the other hand, the goal for students using APLUS is to have their synthetic peers pass the quiz.

To gain an even tighter comparison between learning by teaching and learning by being tutored while controlling the above-mentioned potential confounding, we developed another version of cognitive tutor, APLUS TUTOR. The graphical user interface of APLUS TUTOR is essentially identical to APLUS. APLUS TUTOR requires students to pass the quiz by themselves while allowing them to select and enter practice problems by themselves. As a cognitive tutor, APLUS TUTOR provides the adaptive scaffolding (while students are solving a problem) and adaptive problem selection as mentioned above. It also provides metacognitive scaffolding as APLUS does.

The goal of student using APLUS TUTOR is to pass the quiz by themselves. Students using APLUS TUTOR therefore must select problems by themselves to practice solving equations to pass the quiz. We call this type of learning *goal-oriented practice*. For the sake of explanation, we call the direct instruction driven by COGTUTOR+, *Cognitive Tutoring* (though, APLUS TUTOR also technically provides cognitive tutoring).

In summary, in the current study, we compare three learning strategies—Learning by Teaching (APLUS) that provides metacognitive scaffolding, Cognitive Tutoring (COGTUTOR+) that provides cognitive scaffolding (immediate feedback and just-in-time hint), and Goal-Oriented Practice (APLUS TUTOR) that provides both cognitive and metacognitive scaffolding. In all three versions of interventions, the meta-tutor was visually presented and provided students with cognitive and/or metacognitive scaffolding.

Related Works

There has been essentially three types of teachable agents (TAs) developed so far. The first type of TA is equipped with a genuine machine learning technique (e.g., Matsuda et al. 2011; Michie et al. 1989)—we shall call this type of TA the knowledge-learning teachable agent. The second type of TA adaptively controls pre-compiled knowledge to imitate the performance improvement over time (e.g., Lenat and Durlach 2014; Pareto 2014)—the knowledge-tracing teachable agent. The third type of TA can interpret the subject matter knowledge (e.g., a concept map) that students instructed and utilize the knowledge to solve problems (e.g., Biswas et al. 2005; Zhao et al. 2012)—the knowledge-sharing teachable agent. From students’ point of view, all three types of teachable agents are capable of “learning” skills and knowledge to solve target problems through tutoring interactions.

Despite a long history of learning by teaching in education literature, there have been only a few studies with TA to investigate the theory of tutor learning. Below, we review only those studies that involved an evaluation with actual students. As far as the authors are aware of, there are 7 such studies. Only two of them involve a meta-tutor whereas others did not. Of those 7, we first review 5 studies that do not involve a meta-tutor.

Linear Kid Is for high school math that allows students solve math problems collaboratively with a TA (Jun 2003). In an evaluation study with 28 students, the impact of students' prior competency on tutor learning was measured. In an evaluation study, two groups of students were compared—low vs. intermediate prior competency. It was found that, after using the intervention, the intermediate students outperformed the low students on the accuracy of solving equations, while there was no condition difference in the accuracy of explaining the process of solving equations.

DynaLearn Is an online learning environment that allows students to teach a TA to learn scientific knowledge by manipulating diagrammatic representations, aka a concept map that represent causal and conditional relations (Bredeweg et al. 2009). In addition to the TA, the learning environment involves other types of agents including a *teacher agent* that provides assistance on what, how, and why questions; a *recommendation agent* that provides feedback on a concept map that a student made in comparison to the one made by an expert; a *diagnosis agent* that provides feedback on the results of running the concept map; and a *quiz master* that provides feedback on the quiz that TA takes. In one evaluation study in a graduate level class to learn complex systems, a contribution of a particular question-format that TA asks to students' causal reasoning was examined. Overall, no significant treatment effect was observed. There was no evidence that scaffolding given by agents improved the quality of the concept maps students made. The authors concluded that the short-term intervention is insufficient to affect the model construction process in a significant manner (Mioduser et al. 2012).

Motivated Teachable Agent (MTV) Has a model of intrinsic motivation for learning science lessons to arouses students' interests in learning (Zhao et al. 2012). MTV is embedded into a 3D virtual learning environment that provides primary and secondary school students in Singapore with a culturally familiar scenario to learn science lessons (e.g. transport in living things). In an evaluation study, learning by teaching MTV was compared with regular classroom instruction while the time on task was controlled. The result showed an overall improvement of the test scores from pre- to post-tests, but no condition difference was identified.

TAAG Is a teachable agent for the elementary-school level arithmetic. Students are engaged in games that require arithmetic knowledge to solve—e.g., finding a pair of numbers to match their sum to a given number. The teachable agent asks multiple-choice questions about arithmetic knowledge and strategies to win the game. A quasi-controlled field study (Pareto 2014) showed that while there was a main effect of the treatment for learning conceptual knowledge, which was one of the four constructs, the effect did not appear for other constructs—computing skills, strategy, and other skills related to the task. Although, the AT has a capability to ask questions to students, the impact of question asking on students' learning was no reported.

LECOBA Acts as a learning companion that learns Binary Boolean Algebra (Uresti and du Boulay 2004). Students teach a TA the preconditions under which a particular theorem should apply and the prioritization among conflicting theorems. A meta-tutor

(called the tutor-agent) is incorporated into the learning environment and provides summative comments upon a completion of a solution. The evaluation study was a 2×2 randomized controlled trial where a student's motivation factor was crossed with a TA's competency factor—. There were two levels of motivation settings—in the motivated condition, students were scored based on their tutoring activities hence driven by the extrinsic motivation to score high, whereas in the free condition, students were only encouraged to teach their synthetic peers. There were two levels of competency for the TA—a weak teachable agent vs. a strong teachable agent. The result showed an overall improvement in students' test scores from pre- to post-test, but no condition difference was identified. The effect of meta-tutor for learning by teaching was not measured.

Now, we introduce two studies that involve measuring the effect of a meta-tutor. A pioneering example of TA with a meta-tutor is *Betty's Brain* (Biswas et al. 2005). It is a knowledge-sharing teachable agent that helps students learn causal relations in the ecosystem (e.g., a river system). Betty's Brain is one of the few TAs that have an intensive record of field studies in authentic learning environments. Betty's Brain has been used to investigate various factors of learning by teaching including, for example, effective measures, adaptive scaffolding, and social factors (Biswas et al. 2016). A meta-tutor (called the *mentor agent*) was introduced to the Betty's Brain learning environment from an early stage of the project (Tan et al. 2006).

In one study using Betty's Brain, an impact on students' self-regulated skills on tutor learning was investigated while the role of the meta-tutor was controlled (Biswas et al. 2010). Three types of interventions were compared: Learning by Teaching (LBT) where students learn by teaching Betty's Brain while a meta-tutor agent provided students with corrective feedback on the quiz results and the quality of concept maps students made; Self-Regulated Learning (SRL) where students learn by teaching Betty's Brain while the meta-tutor agent provided students with feedback on self-regulated strategies in addition to the corrective feedback same as LBT; Intelligent Coaching System (ICS) where students learn by being tutored by the meta-tutor that provides corrective feedback (hence no Betty's Brain involved).

The results showed that SRL and LBT students tied on the test scores, but SRL students outperformed LBT on the accuracy of the concept map they created. SRL and LBT outperformed ICS on both the test scores and the map accuracy. A further analysis revealed a hint on the effect of the metacognitive scaffolding while students were teaching Betty's Brain—SRL students (who received metacognitive strategy feedback from the meta-tutor) committed to more advanced and focused monitoring behaviors than LBT students (Biswas et al. 2010; Roscoe et al. 2013). An earlier study on Betty's Brain also reported that metacognitive feedback made students ask more while editing maps and reviewing resources.

The Intelligent Coaching System can be seen as a type of direct instruction that is similar to Goal-Oriented Practice in the current study, where students practice solving problems to pass the quiz by themselves while receiving correct feedback from the system. It is therefore interesting to see if the same effect, i.e., Learning by Teaching outperformed this type of direct instruction even when the domain is different—causal relations vs. equation solving.

A second example of TA with a meta-tutor is *SimStudent*, which is the knowledge-learning teachable agent used in the current paper. SimStudent has been field-tested with more than 2000 middle school students to learn to solve linear algebra equations

(Matsuda et al. 2013b). As described earlier, our previous study showed that metacognitive scaffolding provided by a meta-tutor facilitated tutor learning, but cognitive scaffolding did not. In another study, we tested the effect of self-explanation while students were teaching the teachable agent. SimStudent has a capability to ask “why” questions to solicit students’ self-explanations about their tutoring decisions. For example, if a student provides negative feedback on a step that SimStudent suggested, SimStudent may ask why the student disagreed with the step it suggested. In two school studies conducted in two different years, we compared learning by teaching SimStudent with and without asking “why” questions. The results showed that, in both studies, there was a statistically significant correlation between the amount of self-explanations given and the student’s learning gain (Matsuda et al. 2013b). Therefore, asking “why” questions becomes a permanent feature of SimStudent and used in the current study as well.

In sum, while it seems to be evident that students need adaptive scaffolding to yield attested effect of tutor learning, the current literature has yet to accumulate knowledge from empirical studies on the role of metacognitive scaffolding provided by the meta-tutor. Without knowing how to facilitate tutor learning, the teachable agent technology will not be advanced despite its promising potential for a broader dissemination in authentic learning environments. It is therefore crucial to develop a theory on how the meta-tutor facilitates tutor learning.

Research Questions and Hypotheses

The current paper focus on the relative effectiveness of Learning by Teaching with metacognitive scaffolding to the two versions of learning by being tutored—Cognitive Tutoring (that, by definition, does not provide metacognitive scaffolding) and Goal-Oriented Practice (that provides metacognitive scaffolding). In particular, we will investigate the following specific research questions:

- (Q1) Is Learning by Teaching with metacognitive scaffolding effective for students with low prior competency?
- (Q2) Does Learning by Teaching with metacognitive scaffolding help students learn algebraic conceptual understanding?
- (Q3a) Is Learning by Teaching with metacognitive scaffolding more effective than Cognitive Tutoring?
- (Q3b) How does the effect of learning by being tutored relative to Learning by Teaching change if the adaptive problem selection for mastery learning (Cognitive Tutoring) is replaced with self-paced practice where students select problems by themselves to pass a pre-defined set of quiz problems (Goal-Oriented Practice)?

To answer these research questions, we will test the following specific hypotheses:

- (H1) If the metacognitive scaffolding provides students with hints on problem selection (to teach the teachable agent with more appropriate problems), the timing of the quiz (to quiz the synthetic peer at an appropriate time), and learning resource use, Learning by Teaching with metacognitive scaffolding implemented as APLUS will be effective regardless of the level of student’s prior competency.

- (H2) If the metacognitive scaffolding provides students with hints on reviewing learning resources on algebraic concepts, Learning by Teaching with metacognitive scaffolding (APLUS) will facilitate students' learning on algebraic concepts as well.
- (H3a) Since Learning by Teaching with metacognitive scaffolding is more effective than Learning by Teaching without metacognitive scaffolding, which previously tied with Cognitive Tutoring, Learning by Teaching with metacognitive scaffolding is more effective than Cognitive Tutoring.
- (H3b) Learning by Teaching is as effective as Goal-Oriented Practice when metacognitive scaffolding is available for both conditions.

To test these hypotheses, we conducted two classroom studies in business-as-usual settings where we compared three different learning strategies—Learning by Teaching (APLUS), Goal-Oriented Practice (APLUSTUTOR), and Cognitive Tutoring (COGTUTOR+). In addition to the learning outcome data (i.e., test scores), detailed learning process data (that show interactions between students and an online learning system) were collected. By analyzing the learning outcome data in conjunction with the process data, we will draw conclusions about the effect of Learning by Teaching relative to Cognitive Tutoring and Goal-Oriented Practice.

The next section provides details for these three interventions. The “Classroom in-vivo Evaluation Studies” section then shows the details of the classroom study followed by results and discussion.

Interventions

This section first describes details about each of the three interventions. We then provide a summarization on the similarities and differences among the three interventions.

APLUS: Artificial Peer Learning Environment Using SimStudent

APLUS is an online learning environment where students learn to solve equations by teaching a synthetic peer. Figure 1 shows an example screenshot of APLUS. While details about APLUS have been published elsewhere (Matsuda et al. 2013b), we provide a brief overview of the intervention below.

The synthetic peer is visualized as an avatar in the lower left corner. It is implemented with the SimStudent technology (Matsuda et al. 2015). Prior to using APLUS, students can customize the avatar by changing its hair style, skin color and shirt color. The image of the avatar's face is gender neutral—e.g., Tom in Fig. 1 looks like a male whereas Michelle in Fig. 2 looks like a female, but they use the same image of the face. Students can also name it, e.g., Tom, as shown in Fig. 1. Once students start tutoring SimStudent, they are not allowed to change the avatar image or name.

SimStudent is a machine learning agent that interactively learns skills to solve problems through guided-problem solving—i.e., a student using APLUS acts as a tutor for SimStudent. SimStudent applies inductive logic programming to induce skills in the form of production rules by generalizing given examples. The basic tutoring interactions between a student and SimStudent include the following:

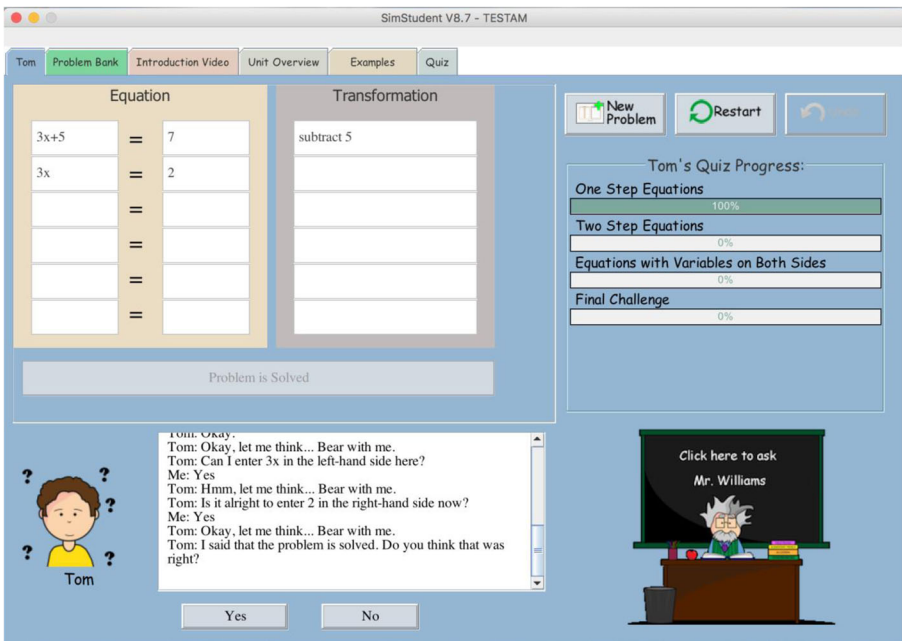


Fig. 1 Example screenshot of APLUS

- (1) A student poses a problem (of their choice from problem bank or one they make up) for SimStudent to solve.
- (2) SimStudent attempts to solve the problem Each step performed by SimStudent is shown on the Tutoring Interface. SimStudent then asks the student for feedback on its correctness. SimStudent is pre-trained on a few one-step equations before students start tutoring so that it has reached a certain level of background knowledge. Thus, SimStudent may initially perform steps both correctly and incorrectly.
- (3) The student provides yes/no feedback on the correctness of the step performed by SimStudent. When the student says “no” to SimStudent’s step, SimStudent then makes another attempt by applying an alternative skill, if any.
- (4) When SimStudent has no skills to apply, SimStudent asks the student for help. The student must demonstrate the step by entering an expression in the Tutoring Interface.
- (5) The student may quiz SimStudent at any time during tutoring by clicking on the [Quiz] button. SimStudent applies productions learned thus far to solve quiz problems as explained below.

The goal for students using APLUS is to have their SimStudent pass the quiz. The quiz has four sections ordered by difficulty—One Step Eq. (1 problem), Two Step Eqs. (2 problems), Equations with Variables on Both Sides (4 problems), and Final Challenge (8 problems which are all equations with variables on both sides). When the [Quiz] button is clicked, SimStudent solves a single problem at a time. When the problem is solved, the individual steps made by SimStudent are displayed along with the

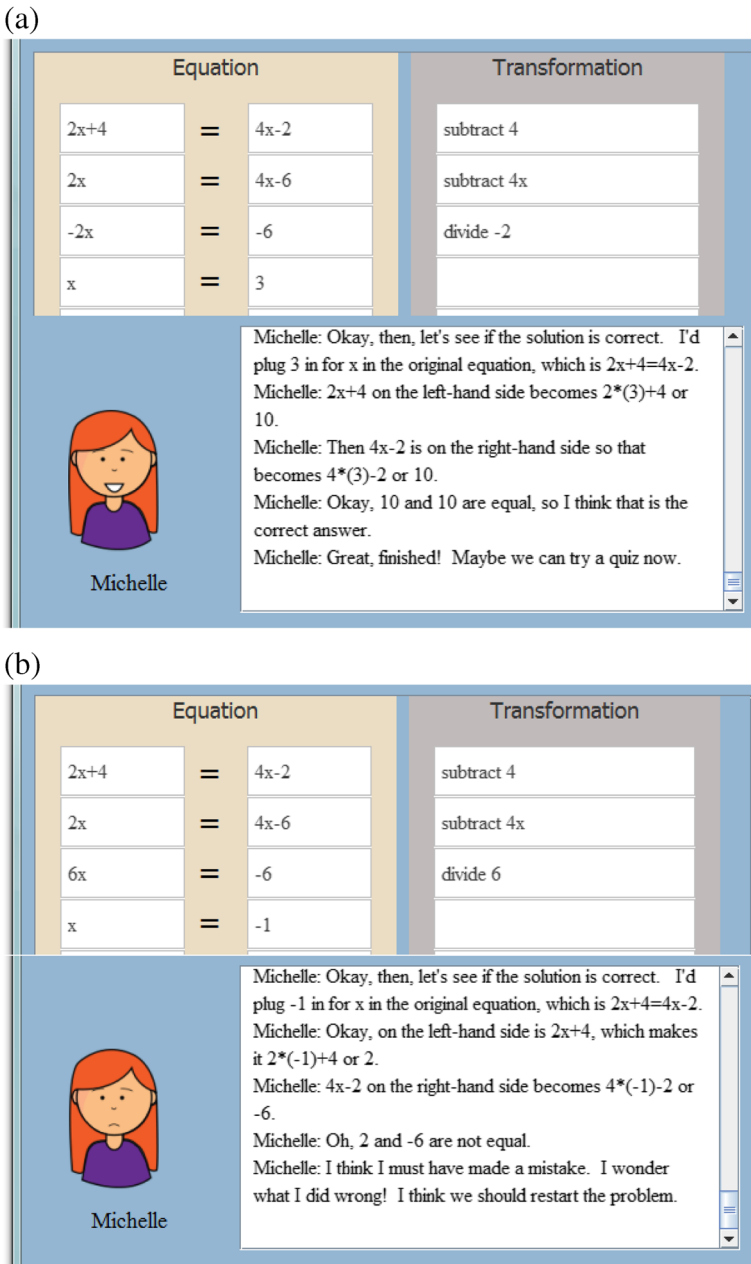


Fig. 2 Sample solution checking dialogues for correct (a) and incorrect (b) solutions

correctness of each step. The Cognitive Tutor Algebra-ITM grades the quiz results (it does not interact with the student—it is used only for quiz and logging purposes). For each section of the quiz, SimStudent must complete the section by solving all problems correctly to proceed to the next section. To “pass” the quiz, SimStudent must complete all quiz sections. Students therefore

must teach SimStudent sufficient skills to solve equations with variables on both sides to achieve the goal.

When a problem is completed, SimStudent occasionally checks a solution by plugging the solution into the original equation and seeing if the equation balances. The *solution checking* happens randomly with 30% of chance. This function was introduced, because we observed in previous studies that students taught SimStudent incorrectly (hence led to an incorrect solution) without knowing they made a mistake. By checking the solution, SimStudent can at least bring an error to student's attention. The solution checking is implemented as a think-aloud monologue done by SimStudent. Figure 2 shows sample monologues of solution checking for a correct and an incorrect solution.

There are learning resources available for students to review: (1) Unit Overview that provides a brief overview of how to solve algebra equations, (2) Examples that provide worked-out examples for the target equations, (3) Intro Video that is a brief video explaining how to use APLUS, and (4) Problem Bank that provides a list of suggested equations to be used for teaching.

SimStudent is an instance of programming by demonstration (W. W. Cohen 1998; Lau and Weld 1998) with inductive logic programming (Muggleton and de Raedt 1994), version space (Mitchell 1982), and iterative deepening search (Russell and Norvig 2003). SimStudent generates a set of *production rules* (where each rule represents a skill) as hypotheses that explain the positive and negative examples of various skill applications. When students teach SimStudent on APLUS, positive and negative examples are given to SimStudent as a combination of feedback and hints provided by the student. The affirmative feedback (i.e., “yes”) and hints become positive examples, whereas negative feedback (i.e., “no”) becomes negative examples (Matsuda et al. 2005). For the current study, SimStudent must learn nine skills to pass the quiz—four skills for steps under “Transformation” (see the Tutoring Interface in Fig. 1), which are to add a same term to both sides, subtract a same term from both sides, multiply both sides with a same term, and divide both sides by a same term; four other skills to actually do arithmetic for steps under “Equation”; and one skill to notice that a problem is solved. For example, for an equation “ $3x+5 = 7$ ”, the first skill to be applied is to subtract 5 from both sides (to enter “subtract 5” in Transformation), and the second skill is to subtract 5 from $3x + 5$ (to enter “ $3x$ ” in Equation).

APLUS includes a teacher agent (aka a *meta-tutor*), called Mr. Williams, as visualized with an avatar on the lower right corner of the APLUS interface (Fig. 1). Unlike the SimStudent avatar, the Mr. Williams' avatar cannot be changed, and it always appears as the one shown in Fig. 1. Mr. Williams provides students with help on how to appropriately tutor SimStudent (called *metacognitive tutoring help*) for the following five metacognitive skills of tutoring that have been identified to be the most troublesome for students in our previous studies:

- (1) Selecting an appropriate next problem to teach to SimStudent—Mr. Williams suggests the student teach a problem from the quiz that SimStudent failed to solve. An example of a help message from Mr. Williams for the problem selection reads “*I see that Tom failed all quiz items. Tom can do better with some practice on two step equations.*”

- (2) Administering the quiz at an appropriate time—After teaching SimStudent, Mr. Williams suggests the student administer the quiz to SimStudent. An example of a help message from Mr. Williams for the quiz reads “*I see Tom passed this quiz section, it might be helpful to understand what else your student knows. So, it would be a good idea to quiz Tom again now.*”
- (3) Reviewing resources, e.g., unit overview and examples, at an appropriate time—When SimStudent does not make any progress on the quiz, Mr. Williams suggests the student review the resources. An example of a help message from Mr. Williams for reviewing resources reads “*I think it’s a good idea to go through the Examples - see the tab above. Make sure you understand all the examples.*”
- (4) Providing Feedback—When SimStudent is asking the student for feedback on the step it performed, Mr. Williams suggests the student provide yes/no feedback. An example of a help message from Mr. Williams for providing feedback reads “*Tom is asking you to justify your answer. You should answer Tom’s question.*”
- (5) Demonstrating a step on which SimStudent gets stuck—When SimStudent is asking the student for help on how to perform the next step, Mr. Williams suggests the student enter a corresponding step in the Tutoring Interface. An example of a help message from Mr. Williams for demonstration reads “*Tom is asking for help. You should tell Tom the next step.*”

The last two types of metacognitive tutoring help are implemented because students are sometimes confused how to use APLUS.

The metacognitive tutoring help is delivered either by request or proactively. Students can click on Mr. Williams anytime to ask their questions about how to teach. When Mr. Williams is clicked, a pop-up menu is then shown with available questions for the student to ask. Mr. Williams also occasionally provides hints proactively (without student’s request). Both for requested and proactive hints, a hint message from Mr. Williams is displayed in a separate dialogue box so that students can perceive it as a private message.

The metacognitive tutoring help, both requested and proactive, is driven by the model-tracing technique (Anderson et al. 1990). We created a *metacognitive model of tutoring* in the form of production rules. There are 19 productions in the metacognitive model of teaching. An example of a production in the metacognitive model of tutoring is for students to review examples when SimStudent failed on the same quiz problem more than three times. To model trace students’ tutoring behavior, we apply a traditional model-tracing technique used for assessing the correctness of student’s solution steps in the cognitive tutor. The system continuously model-traces student’s tutoring activities using the metacognitive model of tutoring. Each time the student makes a tutoring move (which corresponds to an action on the tutoring interface), the system determines if there is a production in the metacognitive model of tutoring that yields the same move. When the system fails to model trace the student’s move, then the system flags a production in the metacognitive model of tutoring that should have been matched with the students’ behavior—i.e., the production that shows an expected behavior. When a particular production has been flagged three or more times, then a hint message for the corresponding production (i.e., metacognitive tutoring help) is proactively given with 60% of chance.

There are two major differences between the versions of APLUS used for our previous study (Matsuda et al. 2011) and the current study—i.e., the introduction of the solution checking and the metacognitive tutoring help. Other than that, the behavior of the two versions of APLUS (from the 2011 study and the current study) is the same including the version of SimStudent, the learning resources available for students to review, the structure and content of the quiz, and the goal of learning (e.g., passing all quiz levels).

CogTutor+

COGTUTOR+ is a cognitive tutor that has the same graphical user interface as APLUS, Fig. 3 shows an example screenshot. COGTUTOR+ provides the student with mastery learning driven by the knowledge tracing technique (Anderson et al. 1990). We designed COGTUTOR+ so that it provides the same adaptive instruction that Carnegie Learning Cognitive Tutor Algebra I™ does—i.e., immediate feedback, just-in-time hint, and adaptive problem selection.

The first two types of adaptive instruction (i.e., immediate feedback and just-in-time hint) are driven by model tracing (Anderson et al. 1992) that compares students' solutions with model solutions. We use the model-tracing engine embedded in Carnegie Learning Cognitive Tutor Algebra I™—COGTUTOR+ is connected to the Cognitive Tutor Algebra I™ through the application programming interface. There are nine skills that are subject to model tracing, all of which are the same skills that SimStudent learns on APLUS. Each step a student enters in the Tutoring Interface of CogTutor+ is colored either red, which indicates that the step is incorrect, or green, which indicates that the step is correct (just like Cognitive Tutor Algebra I™ does). While using COGTUTOR+,

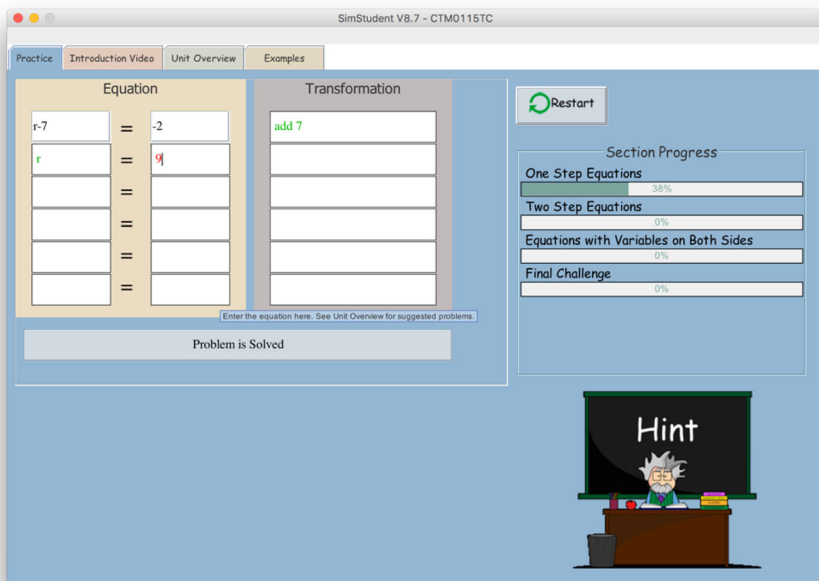


Fig. 3 Example screenshot of CogTutor+

the student can ask for a hint for the next correct step (i.e., just-in-time hint, e.g., “*What should I do next?*”) by clicking on Mr. Williams in the bottom right corner of the screen.

The third type of adaptive instruction (i.e., adaptive problem selection) is driven by knowledge tracing (Corbett and Anderson 1995) that computes the mastery level of individual skills as a probability of applying them correctly. We implemented the Bayesian Knowledge Tracing (BKT) technique. To apply BKT, the initial parameters (mastery, guess, slip, and learning) must be estimated. To make an estimation for the initial parameter valued, we used an existing dataset from DataShop—called “Algebra I 2007-2008 (equation solving units)” available from the project “Algebra I Course”. These data were collected from a school study where participants used Carnegie Learning Cognitive Tutor Algebra I™. We only used data that correspond with the nine skills mentioned above. We applied the contextual estimation method (Baker et al. 2010) to compute the initial parameter values.

The goal for a student using COGTUTOR+ is to achieve a mastery proficiency level for all nine skills across all types of equations that are the same as APLUS—one-step equation, two-step equation, and equations with variables on both sides (as shown in Fig. 3). Student’s progress on the proficiency level is displayed as a bar graph on the right-hand side of the COGTUTOR+ interface. A bar graph shows an average of the proficiency of the skill learning for each quiz level where the “proficiency” is represented as the L parameter of the BKT. For each quiz level, the average L value is computed across skills that are involved in the quiz problems in that level.

We designed COGTUTOR+ to control for the learning resources for students to review with APLUS—i.e., the same Introduction Video, Unit Overview, and Worked-out Examples as APLUS are available. However, there is no metacognitive help provided by CogTutor+ to suggest when students should review these resources.

AplusTutor

APLUSTUTOR is a cognitive tutor that provides the same adaptive instruction as COGTUTOR+, i.e., immediate feedback and just-in-time hint. APLUSTUTOR uses the same model tracing back-end in Cognitive Tutor Algebra I™ as COGTUTOR+. The same 9 skills used for APLUS and COGTUTOR+ as mentioned earlier are used for APLUSTUTOR for model tracing. In other words, the set of skills are controlled across all three versions of systems used in the current study.

Unlike COGTUTOR+, knowledge tracing is not implemented in APLUSTUTOR, i.e., the adaptive problem selection is not provided by APLUSTUTOR. Instead, students choose problems from the Problem Bank or make them up and enter them into the Tutoring Interface by themselves. The lack of knowledge tracing implies that APLUSTUTOR does not compute student’s mastery level, which further implies that the learning goal is not achieving a mastery proficiency. Instead, the goal for students using APLUSTUTOR is to solve all quiz problems correctly by themselves. The quiz sections in APLUSTUTOR are organized in the same way as APLUS. The student may click on the [Quiz] tab to take a quiz at any time. The student is asked to submit a solution for one quiz problem at a time (just like SimStudent solves a single quiz problem at a time), and the system provides feedback on the correctness of the solution. The student can modify an

incorrect solution and resubmit as many times as they want (even without practicing on a problem with cognitive tutoring).

The interface of APLUS TUTOR is almost identical to APLUS except there is no synthetic peer present (Fig. 4). A student enters a problem in the interface and then the cognitive-tutor backend provides the adaptive scaffolding (i.e., immediate feedback and just-in-time hint) while the student is solving the problem. Likewise, APLUS and COG TUTOR+, students may click on Mr. Williams anytime to ask for a hint. In addition to the just-in-time hint (on how to solve a problem), Mr. Williams also provides the following three types of metacognitive hints that are equivalent to the metacognitive tutoring help provided by APLUS: (1) Selecting an appropriate next problem to practice. (2) Taking the quiz at an appropriate time. (3) Reviewing resources. Like APLUS, these types of metacognitive hints are delivered either upon students' request or proactively by Mr. Williams.

Comparison Among Three Interventions

The three online learning systems mentioned above provide different learning opportunities, though all three systems have the same learning objective—i.e., learning to solve three types of equations that is, one-step equations, two-step equations, and equations with variables on both sides. We suppose that for each system, the learning opportunities shown in Table 1 most essentially influence students' learning as described below.

All three conditions have learning resources—i.e., Unit Overview, Intro Video, and Examples. The Problem Bank is available only for APLUS and APLUS TUTOR. For

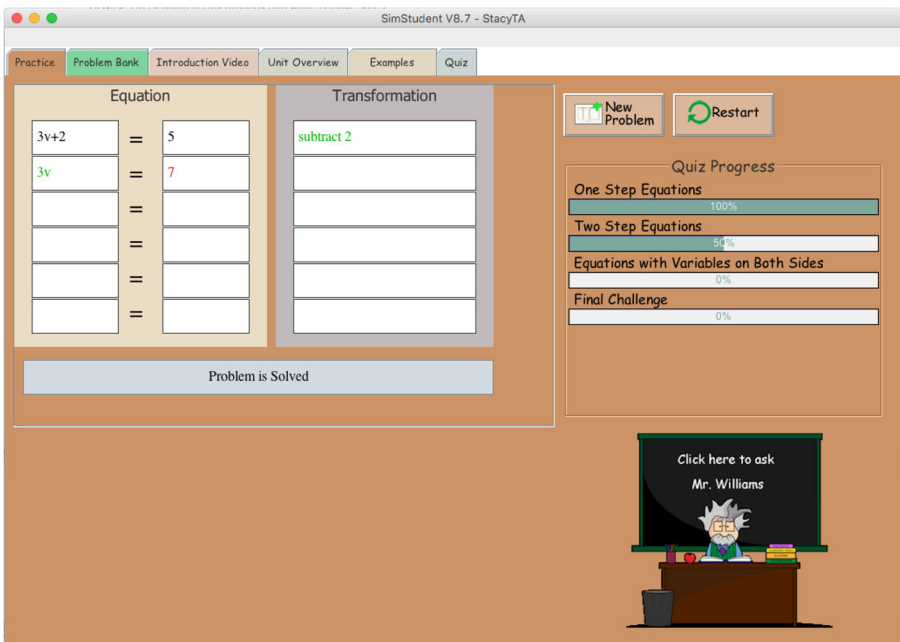


Fig. 4 Example of screenshot of APLUS TUTOR

APLUS, students teach the synthetic peer where they provide immediate feedback and just-in-time hint to their synthetic peer by themselves. On the other hand, for APLUS TUTOR and COGTUTOR+, students practice solving equations through the direct adaptive instruction (aka, cognitive help)—i.e., immediate feedback and just-in-time hint—provided by an embedded cognitive tutor.

For APLUS the synthetic peer takes the quiz, whereas for APLUS TUTOR students take the quiz. Both APLUS and APLUS TUTOR provide a review for the quiz. Passing the quiz is the criteria for completion. COGTUTOR+ does not have a quiz. Instead it provides mastery practice.

Both APLUS and APLUS TUTOR provide metacognitive help on problem selection, timing of quiz, and resource review. In addition, APLUS also provides metacognitive help on feedback and step demonstration.

Classroom Evaluation Studies

To test the specific hypotheses discussed in section “Research Questions and Hypotheses,” we conducted two evaluation studies in authentic business-as-usual classroom settings.

Method and Participants

The two evaluation studies were held in 2016 and 2017, both in late spring. For the 2016 study, two public schools participated with a total of 184 7th and 8th grade students in 12 algebra classrooms. For the 2017 study, one public school participated with a total of 260 6th and 7th grade students in 12 algebra classrooms.

Table 1 The most essential learning opportunities in each online learning system. (*) Guided problem-solving means the adaptive instruction driven by the cognitive tutor, i.e., immediate feedback and just-in-time hint

APLUS (LBT)	APLUS TUTOR (GOP)	COGTUTOR + (CT)
Reviewing learning resources	Reviewing learning resources	Reviewing learning resources
Teaching the synthetic peer (i.e., students provide cognitive help to their peer)	Guided problem-solving (i.e., students receive cognitive help from the cognitive tutor)	Guided problem-solving (i.e., students receive cognitive help from the cognitive tutor)
Reviewing quiz results (taken by the synthetic peer)	Reviewing quiz results (taken by the student)	
Receiving metacognitive tutoring help (problem selection, quiz, resource review, feedback, and step demonstration)	Receiving metacognitive tutoring help (problem selection, quiz, and resource review)	
Completion upon the teachable agent passing Quiz	Completion upon the student passing Quiz	Completion upon the student reaching to the mastery

Three study conditions were implemented: (1) *Learning by Teaching* (LBT) condition where students used APLUS. The students' goal was to have their synthetic peer pass the quiz. (2) *Cognitive Tutoring* (CT) condition where students used COGTUTOR+. The students' goal was to reach the mastery level in solving eqs. (3) *Goal-Oriented Practice* (GOP) condition where students used APLUS TUTOR. The students' goal was to pass the quiz by themselves.

In the 2016 study, all three conditions were used. However, for the 2017 Study, we decided to only compare LBT and GOP, because the 2016 Study showed that GOP is as effective as LBT and CT even when GOP students did not commit to practicing on solving equations (as shown in the RESULTS section below). Since GOP is a new type of intelligent tutoring system with no adaptive problem selection, we assumed that it was important to replicate the observation that GOP tied with LBT, and wanted to gain more statistical power (by dropping the third condition). GOP is also a tighter control for LBT than CT as mentioned above.

Both evaluation studies were randomized controlled trials based on the within-class randomization—i.e., for each classroom, individual students were randomly assigned in one of the study conditions.

A study session at a school ran for six days, one classroom period per day (45 to 50 min). On the first day, all participants took an online pre-test. On the 2nd through 5th day, participants used a corresponding version of the software. At the beginning of the 2nd day, all participants watched the video explaining how to use the software (for about 6 min). Since the video was embedded in the software, participants were able to watch it again anytime if needed. On the last day, participants took an online post-test that was isomorphic with the pre-test. There were two versions of online tests that were randomly assigned to students as pre- and post-test to counterbalance the version difference—i.e., the half of students took the test version A for pre-test and version B for post-test whereas the other half went the other way. The next section provides details about the tests and other measures.

Measures

We measured learning outcomes and activities. Students' *learning outcomes* were measured using an online test that consisted of two parts: The Procedural Skill Test and the Conceptual Knowledge Test. It is sometimes reported that a standardized test can be less sensitive to learning gains than an assessment specifically designed around particular intervention content (Cook et al. 1986; Rohrbeck et al. 2003). Since the primary purpose of our intervention, APLUS (and its control, APLUS TUTOR and COGTUTOR+), is to learn to solve a particular type of questions (i.e., one- and two-step equations, and equations with variables on both sides), we created our own online tests that are intervention oriented, instead of using an existing standardized test.

The *Procedural Skill Test* (PST) had three sections: (1) The Equation section with 10 equation solving items for the same levels of equations (but actually different problems) as the ones included in the quiz used in APLUS and APLUS TUTOR—2 one-step equations (e.g., $a + 7 = 15$), 2 two-step equations (e.g., $5 - 2p = 10$), and 6 equations with variables on both sides (e.g., $-x + 3 = 2 - 4x$). Students only need to enter a solution (can be an integer, a decimal, or a fraction) into the online test form. This section also had a corresponding paper form for students to show their work. (2) The

Effective Next Step section with 2 equation problems that were half solved (e.g., an equation “ $8x+5 = 5x+7$ ” was transformed into an equation “ $3x+5 = 7$ ” by “subtracting $5x$ from both sides”) and four operations were proposed for a next step (e.g., “add 3 to both sides,” “subtract 5 from both sides,” etc.). Students were then asked to identify the correctness by selecting a “yes/no” option of each proposed operation. (3) An Error Detection section with 3 equation problems that were solved incorrectly with intermediate solution steps shown (e.g., an equation “ $10w + 1 = 6 - 4w$ ” was solved with 3 steps; (i) $10w = 5 - 4w$, (ii) $6w = 5$, (3) $w = 5/6$). Students were asked to identify the incorrect step and explain their reasoning.

The *Conceptual Knowledge Test* (CKT) consisted of 24 true/false questions about basic algebra vocabulary—6 items asking about variable terms (e.g., In $3 = 4 - 5b$, 3 is a variable term in the equation. True or false?), 6 about constant terms, 6 about like terms (e.g., $3d$ is a like term for $7a$. True or false?), and 6 about equivalent terms.

As mentioned earlier, there were two isomorphic versions of the online tests. The two versions are identical in their structures and corresponding pair of problems can be solved with the same skills—roughly speaking, they only differ by the numbers and variable letters used.

For each question item both in the PST and CKT, students were encouraged to select the “I don’t know” option when they were not certain about the answer. We introduced this option to discourage students from making a non-educated guess. In our past classroom studies where the same PST and CKT tests were used, the reliability of the online test (Cronbach’s alpha) ranged from 0.76 to 0.84 depending on a test version and a school (Matsuda et al. 2011). For the 2017 study, the Cronbach’s alpha was 0.87 and 0.89 for pre- and post-test respectively.

Students’ learning activity was measured using *learning process data* that showed detailed interactions between individual students and the system. The interactions were automatically collected by the system including problems used for tutoring or practice, solutions entered by the student and the synthetic peer, quiz progress, hint requested, etc. In all three versions of the system, there was a Cognitive Tutor embedded for a logging sake. The correctness of each step made by students and the synthetic peer was judged by the Cognitive Tutor and logged.

Analysis

To test hypotheses mentioned earlier (H1, H2, H3a, and H3b), we evaluated how the intervention affected students’ test scores. We started with applying a repeated-measures ANOVA for each measure (i.e., Equation, PST, and CKT), with test score as the dependent variable, and test-time (which is the *timing* of the test, i.e., pre- vs. post-test) and condition (LBT vs. GOP vs. CT) as independent variables.

The aptitude-treatment interaction (ATI) was tested by splitting into three groups based on their prior competency measured by pre-test scores followed by a two-way ANCOVA with post-test as a dependent variable and condition (LBT vs. GOP vs. CT) and prior competency (Low vs. Mid vs. High) as the independent variables while taking pre-test as a covariate.

We also analyzed learning process data to better understand how students in different conditions interacted with the given system and how the difference in the interaction yielded different learning outcomes. We conducted basic descriptive statistics such as

counting the number of problems practiced, hint messages received, quiz submitted, and learning resources reviewed. We then conducted correlational and regression analyses to understand if and how some of those descriptive statistics were relate with learning.

Results

For the analysis below, we only include students who took both the pre- and post-tests and attended class while using the intervention for at least 3 (out of 4) days. For the 2016 Study, there were 84 students (out of 184 in two schools) who met these criteria—123 students took the pre-test and 152 took the post-test; 114 students met the attendance criteria. Among these students, we excluded 17 ceiling students who scored 100% correct on the pre-test in the Equation section of the Procedural Skill Test. As a result, there are 67 students in the following analysis: 24 in Learning by Teaching (LBT), 22 in Goal-Oriented Practice (GOP), and 21 in Cognitive Tutoring (CT). For the 2017 Study, 3 ceiling students were excluded who scored 100% on the pre-test in the Equation section. There are 141 students in the following analysis: 71 in LBT and 70 in GOP.

For the 2017 Study, there was a technical issue during the post-test that 75 students (which is 54% of the students who met the inclusion criteria) could not open the online test form. All 75 students were 7th graders. As a consequence, for the 2017 Study, only 66 qualified students had PST and CKT test scores and were mostly 6th graders (whereas students in the 2016 Study are all 7th and 8th graders). There was no notable difference in the number of students in each condition who did and did not take the online test form: for 75 who did not take the online test form, 36 were in GOP and 39 were in LBT; whereas for 66 who took the online test form, 34 were in GOP and 32 were in LBT.

Although students could not use the online test form, they were able to work on the Equation section, because this section had a paper form to show their work. Those who used the online test form were also asked to show their work on the paper form and only enter the solution (e.g., $x = 8$) in the online form. We therefore anticipate that the media difference (paper vs. online) made no significant influence of on students' performance. As a consequence, in the following analysis, we only show the Equation section (which is the most essential part of the PST) for the 2017 Study. The Equation section is also shown for the 2016 Study as a comparison.

Learning Outcomes

Table 2 shows the average test scores for both studies comparing the pre- and post-test across study conditions. For the Equation section (of PST), the two studies show the same pattern that test-time (pre vs. post) was a main effect (2016- $F(1,64) = 10.52$, $p < 0.01$; 2017- $F(1,139) = 45.91$, $p < 0.001$), but there was no condition difference (2016- $F(1,64) = 1.27$, $p = 0.77$; 2017- $F(1,139) = 3.48$, $p = 0.06$). For the overall PST (which is available only for the 2016 Study), test-time was also a main effect ($F(1,64) = 15.43$, $p < 0.001$), but no condition difference was detected ($F(2,64) = 0.56$, $p = 0.58$). For CKT (only for the 2016 Study), neither test-time ($F(1,64) = 2.36$, $p = 0.13$) nor

Table 2 Test scores for the 2016 Study (a) and the 2017 Study (b). A number in parentheses shows a standard deviation

(a) The 2016 Study									
	Equation			PST			CKT		
	Pre	Post	d	Pre	Post	d	Pre	Post	d
LBT ($N=24$)	.46 (.32)	.55 (.25)	.28	.55 (.27)	.62 (.22)	.26	.49 (.23)	.53 (.22)	.17
GOP ($N=22$)	.46 (.31)	.50 (.31)	.13	.48 (.25)	.56 (.20)	.32	.48 (.21)	.51 (.14)	.14
CT ($N=21$)	.45 (.32)	.54 (.29)	.28	.54 (.22)	.62 (.24)	.36	.43 (.23)	.47 (.16)	.17
Total ($N=67$)	.45 ^a (.31)	.53 ^a (.28)	.26	.53 ^b (.24)	.60 ^b (.22)	.29	.47 (.18)	.50 (.18)	.17
(b) The 2017 Study									
	Equation			PST			CKT		
	Pre	Post	d	Pre	Post	d	Pre	Post	d
LBT ($N=71$)	.23 (.24)	.34 (.29)	.46						
GOP ($N=70$)	.24 (.24)	.42 (.32)	.75						
Total ($N=141$)	.23 ^c (.24)	.38 ^c (.31)	.63						

(*): Equation: The equation section of PST. PST: Procedural Skill Test. CKT: Conceptual Knowledge Test. LBT: Learning by Teaching. GOP: Goal-Oriented Practice. CT: Cognitive Tutoring. PST except the Equation section and CKT were not available for the 2017 Study. ^a $F(1, 64) = 10.52, p < 0.01$, ^b $F(1, 64) = 15.43, p < 0.001$, ^c $F(1, 139) = 45.91, p < 0.001$

condition ($F(2,64) = 0.67, p = 0.52$) were main effects, and no interaction between test-time and condition was detected.

In sum, *for learning to solve equations, there was no condition difference, and students showed improvement from pre-test to post-test measured by the PST test.*

To investigate how students' prior competency (measured by the pre-test score) affected the effect of each intervention, we split students into three groups of equal size—Low, Mid, vs. High—based on their pre-test score of the Equation section. Table 3 shows the average post-test score on the Equation section for each condition and prior. Figure 5 is an interaction plot showing the Equation post-test score comparing each condition crossed with the prior. We then ran a two-way ANOVA with the post-test score as the dependent variable, and prior (Low, Mid, and High) and condition (LBT vs. GOP vs. CT for the 2016 Study, whereas LBT vs. GOP for the 2017 Study) as independent variables (in this order). The interaction term (among condition and prior) was not statistically significant for either study; for 2016, $F(4, 58) = 0.76, p = 0.56$; for 2017, $F(2, 135) = 0.16, p = 0.85$; indicating that *there was no aptitude-treatment interaction observed.*

It is worthwhile to notice that numerical definitions of the level of prior (Low, Mid, and High) are slightly varied between 2016 and 2017 Studies as shown in Table 3 as the differences in the mean pre-test scores. Overall, students in the 2017 Study had relatively lower prior than the 2016 students. This implies that *LBT was equally effective as GOP for a wide variety of students in terms of their prior competency.*

In summary, hypothesis H1 was supported. *By adding metacognitive scaffolding, learning by teaching realized by APLUS became effective for students with various prior competencies (measured as the pre-test), in particular even students with low prior competency were benefit from learning by teaching when metacognitive*

Table 3 The average post-test score on the Equation section with a three-way split based on the pre-test score for the 2016 Study (a) and the 2017 Study (b)

	Low (N = 23)			Mid (N = 22)			High (N = 22)		
	Pre	Post	d	Pre	Post	d	Pre	Post	d
(a) The 2016 Study									
LBT (N = 24)	.09 (.08)	.37 (.21)	3.25	.41 (.12)	.51 (.19)	.42	.84 (.07)	.76 (.16)	-1.14
GOP (N = 22)	.13 (.07)	.23 (.19)	1.42	.48 (.10)	.53 (.18)	.50	.81 (.09)	.81 (.20)	0
CT (N = 21)	.10 (.04)	.24 (.21)	3.75	.43 (.12)	.56 (.19)	.08	.82 (.08)	.81 (.07)	.13
Total	.11 ^a (.07)	.27 ^a (.20)	2.29	.44 ^b (.12)	.54 ^b (.18)	.83	.82 (.08)	.80 (.15)	.25
(b) The 2017 Study									
LBT (N = 71)	.02 (.04)	.19 (.22)	4.25	.20 (.05)	.26 (.24)	1.2	.54 (.20)	.62 (.25)	.40
GOP (N = 70)	.01 (.03)	.26 (.26)	1.42	.17 (.06)	.33 (.21)	2.66	.46 (.21)	.63 (.31)	.81
Total	.01 ^c (.04)	.22 ^c (.25)	5.25	.19 ^d (.06)	.30 ^d (.23)	1.83	.50 ^e (.21)	.62 ^e (.28)	.57

^a $F(1, 20) = 11.56, p < 0.01$, ^b $F(1, 19) = 8.77, p < 0.01$ ^c $F(1, 45) = 29.60, p < 0.001, p < 0.01$, ^d $F(1, 45) = 8.76, p < 0.01$, ^e $F(1, 44) = 11.59$

scaffolding was available. The same pattern was observed for the other two conditions. As a consequence, there was no aptitude-treatment interaction observed in the current study.

Hypothesis H2 was not supported. *There was no evidence for the current implementation of learning by teaching with metacognitive scaffolding (APLUS) being effective for learning conceptual knowledge measured with the Conceptual Skill Test for the 2016 Study.*

Hypothesis H3a was not supported. *Learning by Teaching with metacognitive scaffolding was as effective as the traditional Cognitive Tutoring (which did not have metacognitive scaffolding).* H3b, on the other hand, was supported. The current data showed that *Learning by Teaching with metacognitive scaffolding was as effective as Goal-Oriented Practice with metacognitive scaffolding.*

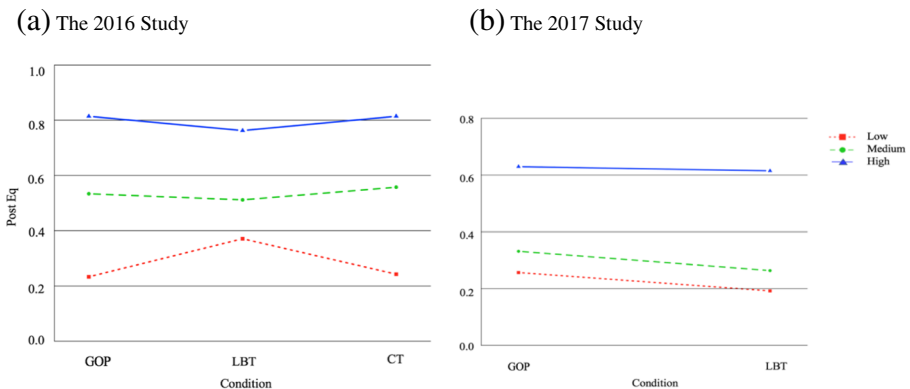


Fig. 5 Interaction plots on the Equation PST post-test score across condition and prior (Low, Medium, and High based on the Equation PST pre-test score) for the 2016 Study (a) and the 2017 Study (b). The ATI was not confirmed for either study

Learning Process

To further understand how the process of learning differs among different study conditions, the learning process data were analyzed.

Learning Activities

The following learning activities were analyzed: (1) the number of times that just-in-time hint was provided by the cognitive tutor (i.e., Mr. Williams), (2) the number of equation problems practiced or tutored, (3) the number of quiz taken and/or reviewed, and (4) the number of times learning resources are reviewed.

Table 4 shows the average frequency of each type of learning activity per individual student. **Problem Entered** shows the number of equation problems entered into the system. The term “enter” is used slightly differently for each learning strategy. For Learning by Teaching (LBT), it corresponds to students entering problems for tutoring their synthetic peer. For Goal-Oriented Practice (GOP), it is about students entering problems for themselves to practice while receiving the adaptive scaffolding from the cognitive tutor. For Cognitive Tutoring (CT), it is about the system posing problems for students to practice while providing them with the adaptive scaffolding. **Cognitive Hint Received** shows the number of cognitive hints students received. Note that no cognitive hint is available for LBT (as indicated as ‘n/a’ in the table). **Quiz** shows the number of times quiz problems were submitted either by SimStudent in LBT or students in GOP. Either case, quiz problems were submitted one at a time. As a reminder, for LBT, when a quiz problem is solved incorrectly, students need to teach SimStudent on more problems and have SimStudent solve the same quiz problem again—simply having SimStudent to redo the quiz without teaching does not change the quiz result at all. For GOP, students may submit the same quiz problem multiple times with different solutions until it is solved correctly. **Resource Review** shows the number of times the four types of learning recourses were reviewed—Unit Overview, Problem Bank, Introduction Video, and Example Solutions. Table 5 shows the breakdown of the frequency count for Resource Review.

To our surprise, Goal-Oriented Practice (GOP) students submitted the quiz a notable amount of times (labeled as Quiz in Table 4) while practicing on a notably small number of problems with cognitive tutor (Problem Entered). A detailed analysis of the process data revealed that *GOP students spent a remarkable amount of time “editing”*

Table 4 Average frequency of each type of learning activities per individual student

	Problem Entered		Cognitive Hints Received		Quiz		Resource Review	
	2016	2017	2016	2017	2016	2017	2016	2017
LBT	21.7	12.8	n/a	n/a	18.0	15.4	22.5	21.0
GOP	7.3	9.6	11.5	16.3	54.7	46.5	15.7	16.3
CT	34.8	-	40.7	-	n/a	-	13.5	-

An ‘n/a’ indicates that corresponding learning activity is not available. The table compares 2016 and 2017 Studies. A ‘-’ indicates that the corresponding condition was not included in the study

Table 5 Average frequency of reviewing each type of resource per individual student

	Unit Overview		Problem Bank		Introduction Video		Example Solution	
	2016	2017	2016	2017	2016	2017	2016	2017
LBT	2.3	3.8	16.0	7.7	2.2	3.5	2.0	6.0
GOP	2.3	2.6	5.6	3.4	2.1	2.4	5.6	7.9
CT	1.3	–	1.1	–	1.9	–	9.1	–

The table compares the 2016 and 2017 study. A ‘-’ indicates that the corresponding condition was not included in the study

and re-submitting their quiz solutions, rather than entering problems to the system and practicing on them with cognitive tutoring. Since the system provides corrective feedback on quiz solutions, students knew which step was wrong. It is therefore likely that they simply made modifications on the incorrect steps and submitted the “edited” solution. For the 2016 Study, the average total number of times individual students submitted a quiz problem is 54.7 ± 41.3 (which is 4.8 ± 6.6 submits per quiz problem). For the 2017 Study, it was 46.5 ± 30.3 (5.3 ± 6.7 submissions per quiz problem). Students seem to have learned to solve equations on a trial-and-error basis by simply “editing” quiz solutions based on the feedback from the system and resubmitting until they pass the quiz. We shall call this curious style of learning, *Learning by Editing*. Quite interestingly, even though Learning by Editing could be considered as “gaming” the system (Baker et al. 2008), GOP students, on average, achieved the same proficiency level as students in other conditions.

Not surprisingly, Cognitive Tutoring (CT) students (who did not work on the quiz and were given problems by the cognitive tutor based on the mastery criteria driven by the Bayesian knowledge tracing) practiced on more problems than students in any other condition.

These patterns appeared repeatedly in both studies. Figure 6 shows the number of problems practiced during the four days of intervention comparing the 2016 and 2017 Studies. The *t*-test revealed that GOP students practiced on fewer problems than LBT students; for the 2016 Study, $M_{LBT} = 22 \pm 8.9$ vs. $M_{GOP} = 7 \pm 7.5$; $t(34) = 5.42$, $p < 0.001$; for the 2017 Study $M_{LBT} = 13 \pm 6.6$ vs.

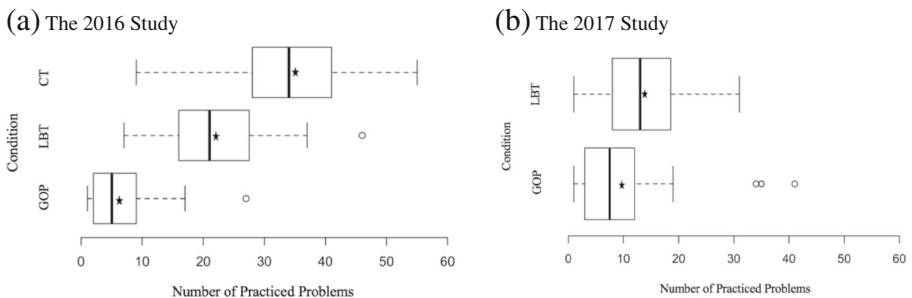


Fig. 6 Boxplots showing the number of practiced problems by students in each condition during four days of intervention comparing the (a) 2016 Study and (b) 2017 Study. An asterisk shows a mean

$M_{\text{GOP}} = 10 \pm 9.4$; $t(81) = 2.07$, $p < 0.05$. Figure 6 shows that CT students in the 2016 Study practiced on the greatest number of problems ($M_{\text{CT}} = 35 \pm 11.1$) among three conditions.

A regression analysis revealed that the number of times quiz submitted was not a reliable predictor of the post-test score when the pre-test score was entered into the model as the primary factor: For the 2016 Study, the pre-test score had a statistically reliable predictive power, $F(1, 19) = 43.93$, $p < 0.0001$; whereas the number of quiz submission did not, $F(1, 19) = 0.03$, $p = 0.87$. For the 2017 Study, the pre-test score was a reliable predictor, $F(1, 66) = 33.05$, $p < 0.0001$; whereas the number of quiz submission was not, $F(1, 66) = 0.04$, $p = 0.84$.

In summary, regarding the number of problems practiced, the CT students were exposed to the most practice problems among the three conditions. The GOP students were exposed to the least practice problems, but instead they submitted and re-submitted the quiz for a notable amount of times. Since GOP students practiced on less than two equations per day on average, it is arguably the case that GOP students learned to solve equations through a quiz cycle—submitting a quiz, receiving corrective feedback, revising (or “editing”) solutions, re-submitting the quiz, and repeat. Most interestingly, *despite the difference in the learning activities among three learning strategies, there was no difference in students’ achievement observed in the current studies*. The number of quiz submissions, however, does not have a statistically reliable predictive power for learning when students’ prior competency (measured as pre-test score) is entered into the model, which suggests that it is not merely the number of times the quiz is submitted that contributes to learning. Further study is necessary to investigate how Learning by Editing contributes to learning.

Quiz Progress

We analyzed how students in Learning by Teaching (LBT) and Goal-Oriented Practice (GOP) made progress on the quiz. Both APLUS (LBT) and APLUS TUTOR (GOP) have four quiz levels (as described in the “Interventions” section). To measure the progress on the quiz, we quantified the quiz levels such that One-step Equation is coded as level 1, Two-step Equation is 2, Equations with Variables on Both Sides is 3, and Final Challenge is 4. In the following analysis, we use these numeric levels.

For the 2016 Study, there were 8 students in Learning by Teaching (LBT) who passed all quiz levels and 18 students in Goal-Oriented Practice (GOP) who passed all quiz levels over the 4 intervention days. For the 2017 Study, 5 and 30 students in LBT and GOP respectively passed all quiz levels.

Figure 7 shows the average quiz level passed on each intervention day for the 2016 Study (a) and the 2017 Study (b). Notice that the GOP condition plots the quiz levels that *students* passed, whereas the LBT condition plots the quiz levels that *teachable agents* passed. The data show that GOP students reached higher quiz levels than LBT students on Day 1. The average highest quiz level passed on Day 4 was 3.0 for LBT and 3.3 for GOP in the 2016 Study whereas 2.4 and 2.8 for LBT and GOP respectively in the 2017 Study. These differences on Day 4 are not statistically significant; for the 2016 Study $t(3.5) = 0.45$, $p = 0.67$; and for the 2017 Study $t(8.6) = 1.20$, $p = 0.26$. However, when the quiz progress is aggregated across all days, the differences were

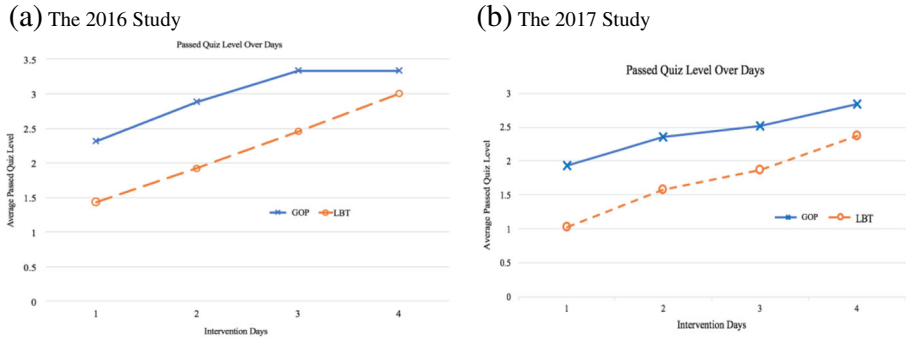


Fig. 7 Transition of quiz levels for Goal-Oriented Practice (GOP) and Learning by Teaching (LBT) for the 2016 Study (a) and 2017 Study (b). The X-axis shows intervention days and the Y-axis shows the average quiz level passed where One-Step Equation is 1, Two-Step Equation is 2, Equation with Variables on Both Sides is 3, Final Challenge (which involves only equations with variables on both sides) is 4

statistically significant; for the 2016 Study $t(79.9) = 3.69, p < 0.001$; and for the 2017 Study $t(230.2) = 10.45, p < 0.001$.

On the other hand, there is no difference in the “rate” of the quiz progress. Linear regression analyses with quiz-level as the dependent variable and intervention-day and condition (GOP vs. LBT) as independent variables did not reveal a difference in the slope (i.e., the “rate” of the quiz progress) among the two conditions in the 2016 nor the 2017 Study.

In summary, *GOP students reached to a higher quiz level quicker than LBT students, but there was no notable condition difference in the “rate” of the quiz progress.* A reason for LBT students starting at a lower quiz level than GOP students on Day 1 is arguably because some of them taught SimStudent incorrectly at the beginning, which screwed up SimStudent’s competency on the quiz and took long time to recover. SimStudent was pre-trained on one-step equation, which means that if LBT students quizzed SimStudent *before* they taught anything at all, SimStudent had passed the first quiz level. Once SimStudent is taught incorrectly and (as a consequence) learns incorrect productions, it often takes a long time for SimStudent to re-learn correct productions. GOP students might have been equally likely to make mistakes on the first quiz level, but it is arguably the case that they re-submitted the failed quiz several times and eventually made it correct, which is quicker than teaching SimStudent. It is interesting, though, that both conditions showed a comparable “rate” of improvement.

Usage of the Metacognitive Scaffolding

To understand how students were exposed to metacognitive scaffolding—i.e., five types of metacognitive tutoring help of APLUS (for LBT) and three types of metacognitive help of APLUS TUTOR (for GOP)—we counted the number of hint messages that individual students received. We are particularly interested in understanding how students requested help by themselves vs. received proactive help that Mr. Williams provided. Table 6 shows the average frequency count of receiving hint messages from Mr. Williams by request vs. proactively, broken down into different types of metacognitive help. Since there is a notable condition difference (LBT vs.

Table 6 The frequency of metacognitive help received per problem by students in each condition in the 2017 Study

		Review Resource	Quiz	Problem	Demonstrate Step	Justify Answer	Total
LBT	Proactive	0.3	0.6 ^a	1.7	0.0	0.0	2.6
	Requested	0.2	0.2	0.6 ^b	0.01	0.03	1.0
							Total: 3.6
GOP	Proactive	0.3	1.5 ^a	2.1	n/a	n/a	3.9
	Requested	0.2	0.2	0.2 ^b	n/a	n/a	0.5
							Total: 4.4

a: $t(102) = 3.81, p < 0.001$

b: $t(95) = 3.77, p < 0.001$

GOP) in the number of problems students practiced as mentioned above, the table shows the average number of hints received *per practice problem* aggregated across students. Although, two types of metacognitive-hint delivery (by request vs. proactive) were available in both the 2016 and 2017 Study, it was only for the 2017 Study that the information about the type of hint delivery was logged. Therefore, Table 6 only shows data from the 2017 Study.

Figure 8 is a visualization of Table 6 with the distribution of each type of metacognitive hint received. Hashed (blue) bars show APLUS TUTOR (GOP) and solid (red) bars show APLUS (LBT).

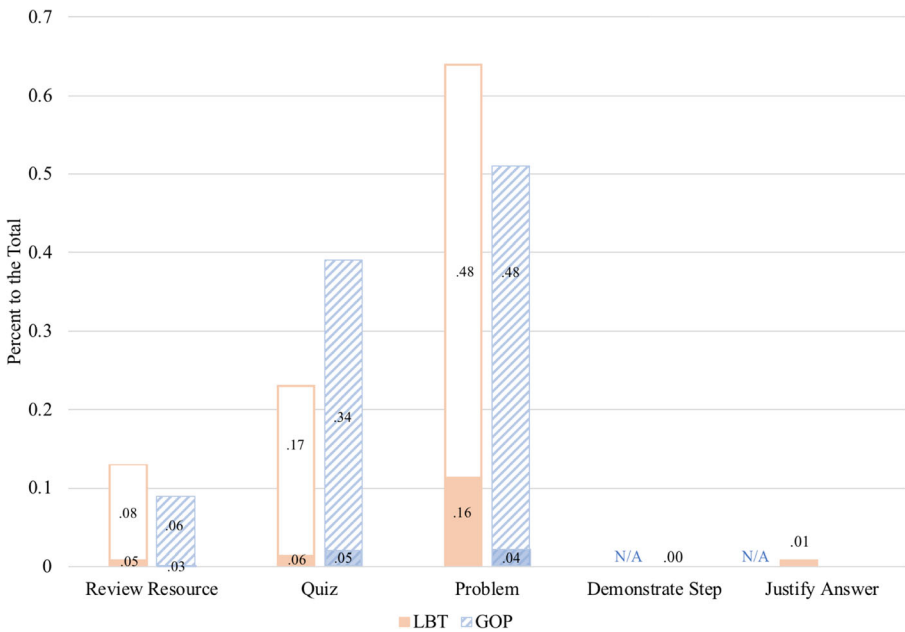


Fig. 8 The distribution of the average amount of hints received by individual students for each type of metacognitive hint represented as a ratio to the total hints received. A solid (red) bar shows APLUS and a hashed (blue) bar APLUS TUTOR. A darker area shows hints requested, whereas a lighter area shows hints provided by Mr. Williams proactively

(red) bars show APLUS (LBT). A darker area shows hints *requested* by students, whereas a lighter area shows hints *proactively provided* by Mr. Williams. The X-axis shows hint type, and the Y-axis shows a transformation of values shown in Table 6 into a ratio relative to the total for each condition—for example, the LBT students received 0.3 hints on Review Resource proactively in average with the total of 3.6 hints received hence the ratio is 0.08.

As the graph shows, students rarely requested help. In total, GOP students received more help *proactively* than LBT students (3.9 vs. 2.6), whereas LBT students *requested* more help than GOP students (1.0 vs. 0.5). Table 6 indicates that these differences are due to the differences in Quiz and Problem—GOP students received more metacognitive helps on Quiz *proactively* than LBT students (1.5 vs. 0.6), whereas LBT students *requested* more metacognitive helps on Problem than GOP students (0.6 vs. 0.2). Both of these differences were statistically significant as shown in the table with t-statistics.

The above observation implies that the LBT students asked Mr. Williams what problem they should teach next on about every other problem whereas the GOP students did so once for every five problems. The LBT students were apparently concerned about selecting appropriate problems for teaching, whereas GOP students rarely practiced on problems.

It is also interesting to see that GOP students received more than twice as many metacognitive helps on Quiz *proactively* than LBT students. Both APLUS (for LBT) and AplusTutor (for GOP) were equipped with the same algorithm to *proactively* provide metacognitive help based on a status of the completion of quizzes and practice problems. The reason that GOP students received more proactive Quiz helps is likely because they completed more quiz levels—when a quiz level is completed, the system occasionally provide a suggestion (i.e., “help”) to proceed to a next quiz level.

There is a general trend shown in the table—students seldom received hints on Demonstrate Step and Justify Answer (these types of help are about how to use the system when teaching the teachable agent and not available for GOP students hence the n/a in the table). Students also very rarely received a help on Review Resources. Given there are students who did not make a steady progress on the quiz, the system should have proactively provided more help on reviewing resources. Fine tuning the timing of the metacognitive help on resource review and evaluate its effect on tutor learning is an important agenda for a future research.

To understand how receiving metacognitive help facilitated students’ learning, a correlation analysis was conducted for post-test score as dependent variable with pre-test (as the first term in the regression model) and the total number of helps received as independent variables. The results show that it is only the pre-test score that has statistically reliable predictive power for the post-test score, and the same trend appeared both for LBT and GOP students. For LBT: Pre-test $F(1, 178) = 93.93$, $p < 0.001$, Number of Helps $F(1, 178) = 0.38$, $p = 0.54$; for GOP: Pre-test $F(1, 178) = 93.93$, $p < 0.001$, Number of Helps $F(1, 178) = 0.38$, $p = 0.54$. As mentioned earlier, since only the 2017 Study had the data about hint type, this analysis was conducted only for the 2017 Study.

To see whether students with different levels of prior competency were exposed to the metacognitive help differently, the frequency count shown in Table 6 was broken down into three groups based on students’ prior competency measured by the pre-test

score. Table 7 shows the breakdown where each cell shows an average number of metacognitive help received per problem aggregated across students. As shown in the table, there was no notable difference in the number of megacognitive help received among students with different level of prior competency for the LBT students. On the other hands, for the GOP students, there is a general trend that the lower group of students received more help on Review Resource and Quiz.

The data shown in Table 7 do not inform much about why we did not see the aptitude-treatment interaction (ATI) between the intervention (LBT vs. GOP) and students' prior (Low vs. Mid vs. High) when the metacognitive hint was available. If the lack of ATI was due to *the amount of metacognitive hint*, then Table 7 might have shown a notable difference between Low, Mid, and High prior students.

In sum, *the current data do not confirm any direct correlation between the amount of metacognitive helps received and learning gain. We also found that a simple count of metacognitive helps received did not explain why a formerly found aptitude-treatment interaction was not present this time.* Further study is needed to investigate how the metacognitive scaffolding facilitated tutor learning.

Discussion

The primary focus of the current study is to compare three learning strategies—one implementation for learning by teaching (APLUS) and two implementations for learning by being tutored (APLUSTUTOR and COGTUTOR+). In particular, the results from two classroom studies showed that by adding metacognitive scaffolding to the online

Table 7: A breakdown of the frequency of metacognitive help received (Table 6) based on students' prior competency (High vs. Mid vs. Low)

		Review Resource	Quiz	Problem	Demonstrate Step	Justify Answer
(a) LBT.						
Low	Proactive	0.3	0.7	2.0	0	0
	Requested	0.1	0.2	0.5	0	0.02
Mid	Proactive	0.3	0.4	1.7	0	0
	Requested	0.2	0.2	0.8	0	0.1
High	Proactive	0.3	0.8	1.6	0	0
	Requested	0.2	0.2	0.5	0.02	0
(b) GOP.						
Low	Proactive	0.5	2.1	1.8	n/a	n/a
	Requested	0.2	0.4	0.4	n/a	n/a
Mid	Proactive	0.2	1.9	2.9	n/a	n/a
	Requested	0.1	0.3	0.1	n/a	n/a
High	Proactive	0.2	1.0	2.0	n/a	n/a
	Requested	0.1	0.1	0.1	n/a	n/a

Table (a) shows students in LBT and (b) in GOP.

learning environment for learning by teaching, a previously observed aptitude treatment interaction (Matsuda et al. 2011) that students with low prior competency benefit more from learning by being tutored disappeared. Regardless of the prior competency, *the current data did not suggest any differences among students in all three learning-strategy conditions for the level of proficiency achieved after using the interventions for four days*. The metacognitive scaffolding might have well helped all levels of students understand how to teach appropriately, which facilitated tutor learning.

Figure 9 depicts this result. In the figure, the y-axis shows whether the metacognitive scaffolding in the online learning environment is available or not. “OFF” shows the results from our previous study (Matsuda et al. 2011) where metacognitive scaffolding was not available, whereas “ON” shows the results from the current study with metacognitive scaffolding.

Other than the availability of the metacognitive scaffolding, the behavior of the interventions was the same—i.e., the same goals, the same learning resources, the same quiz levels, etc. The two bars on the bottom half of the figure show that when metacognitive scaffolding is not available, learning by teaching is less beneficial for students with low prior, but they are equally beneficial for other students. Two bars on the upper half of the figure show that there is no aptitude-treatment interaction when metacognitive scaffolding is available. Along with the results from our previous studies where the availability of cognitive and metacognitive scaffolding was controlled (Matsuda et al. 2016; Matsuda et al. 2014), the current study implies the effect of metacognitive scaffolding for successful Learning by Teaching.

The presence of metacognitive scaffolding for learning by teaching, however, does not affect student’s learning on conceptual knowledge. The current study replicated lessons learned from previous studies (Matsuda et al. 2016; Matsuda et al. 2013b) that

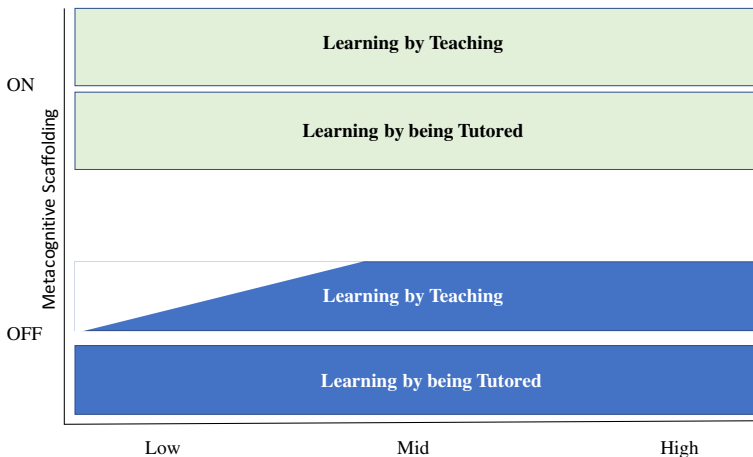


Fig. 9 Comparison of the effect of two learning strategies—learning by teaching vs. learning by being tutored. The x-axis shows student’s prior competency. The y-axis shows whether the metacognitive scaffolding in the online learning environment is available or not. The height of a bar area metaphorically shows an effect of a learning strategy—the higher the bar, the more effective the corresponding learning strategy is. Two bars on the bottom half of the figure show that when metacognitive scaffolding is not available, learning by being tutored is more beneficial for students with low prior, but they are equally beneficial for other students. Two bars on the upper half of the figure show that there is no aptitude-treatment interaction when metacognitive scaffolding is available

the current implementation of APLUS does not necessarily impact students' performance on the Conceptual Knowledge Test regardless of the availability of metacognitive scaffolding. Simply pointing students to the learning resources that contains detailed explanations about conceptual knowledge at an appropriate time did not help. Further investigation is necessary to understand how best we can help students learn conceptual understanding.

Of course, the absence of statistics to reject a null hypothesis, however, does not mean that all three proposed interventions are indeed equally effective or the learning by teaching does not promote learning on conceptual understanding. The current data might reflect a flaw in the validity of the measure used for pre- and pos-tests. The measured reliability of the test (the Cronbach alpha) was reasonably high—0.76–0.89 as mentioned in the section “Measures.” However, those tests might have missed the adequate sensitivity of the latent skills for which the students' learning was actually facilitated by the system.

The difference of the test media (online vs. paper) might have also affected the results. The current data suggest that students who were able to “correctly” teach the teachable agent during the intervention period failed to solve the same type of equations correctly on the post test. We speculate that this is arguably due to a difference between recognition and production. While teaching the teachable agent, students often provide yes/no feedback to the steps made by the teachable agent. Providing the yes/no feedback is supposedly easier for students to do than suggesting a next step by themselves, which is what students needed to do on the test. Students might have learned a skill to recognize correct steps that might not necessary increase the level of proficiency in solving equations.

Alternatively, it might be the case that the proposed interventions were in fact equally effective. To investigate if this is the case, an additional study needs to be conducted with different measures (i.e., pre- and post-test) to see if the results are replicated.

The current paper explored the similarities and dissimilarities among learning by teaching and learning by being tutored. First, we found a notable difference in the amount of problems students practiced while achieving the same level of learning gain. Cognitive Tutoring (COGTUTOR+) students needed to practice on 60% more problems than Learning by Teaching students. However, the comparison for the amount of problems practiced by students in three different learning strategies needs some caution. We must consider the fact that Goal-Oriented Practice (APLUSTUTOR) students apparently learned quite a lot from editing and re-submitting the quiz with the feedback from the system (in addition to the small number of practice problems). Learning by Teaching (APLUS) students, on the other hand, might have learned by teaching on relatively a larger number of problems (in addition to observing the teachable agent solving quiz problems). Furthermore, during the 2016 Study where COGTUTOR+ was used, we received anecdotal input from students expressing their discontent that the tutor insisted them to continue practicing on excessive number of problems, which implies that the number of problems practiced for CogTutor+ reported in the current paper might be unnecessarily inflated.

Nonetheless, since the time on task was controlled for both studies (all students spent the same amount of time on the intervention in the classroom), it is arguably fair to say that *all three learning strategies used in the two studies require a relatively equal*

amount of time to achieve the same proficiency level. Future studies must be conducted to replicate the results to clarify potential confounding due to the system implementation.

Second, it is interesting to see that two different learning goals (a) students passing the quiz by themselves (Goal-Oriented Practice with APLUS TUTOR) and (b) having the synthetic peer pass the quiz (Learning by Teaching with APLUS) have the equal impact on *student's* learning outcome. This result somewhat contradicts a previous finding on the ego protective buffer (Chase et al. 2009)—students' learning is facilitated when they perceive a third party agent (e.g., a synthetic peer) as the target for blame of a failure, i.e., "*it is the synthetic peer that failed on the quiz, not me!*" Further investigations are necessary to understand why we do not observe this phenomenon.

Third, yet another interesting observation, though rather serendipitous, is that Goal-Oriented Practice (GOP) students primarily focused on editing quiz solutions with corrective feedback from the system and re-submitted them repeatedly until passing the quiz (aka Learning by Editing as discussed earlier). This peculiar students' tendency is reasonable (though it is an after the fact), given that GOP students' goal is to pass the quiz. GOP students might have wanted to be "done" many times just as they like to win a game. Learning by Teaching (LBT) students who did not actually take the quiz (but their agents did) do not have this pleasure. Therefore, we argue to consider this behavior as a nature of GOP (as opposed to a confounding factor). What is striking us is that this Learning by Editing strategy lead students to an equal level of learning as the other two learning strategies. The underlying cognitive mechanism of learning-by-editing must be explored in the future. One might argue that GOP students were gaming the system. There was no data collected this time to determine if students were gaming or not. Therefore, we cannot really draw any inference about the impact of gaming the system on tutor learning. The potential of gaming must be addressed in future studies, perhaps by integrating a technology to detect a moment of gaming.

The current data suggest a very little about students' motivation and their engagement. Yet, the data have some indications that Learning by Teaching (LBT) students in the current study were indeed engaged in tutoring their synthetic peers—e.g., consistent improvement on the agent's performance measured as the progress on quiz, relatively frequent *request* for help on problem selection, and a fair amount of problems actually tutored. The anecdotal observations during classroom study indicate that LBT students were very excited about themselves teaching a computer agent and watching their synthetic peer taking a quiz (successfully, in particular). LBT has a potential to externally motivate students—e.g., by adding motivative and attractive avatars (e.g., Bredeweg et al. 2013; Zhao, and Ailiya, and Shen, Z. 2012). Yet, in the current literature, there is a lack of knowledge to the connection between the behavioral characteristics of the teachable agent and the tutor learning outcome. Further study is needed to investigate the theory of students' motivation when learning by teaching and its implication to tutor learning.

Related to the issue of gaming, there is a concern regarding the current design of Goal-Oriented Practice about "shallow" learning that by definition allows students to learn skills to solve equation based on surface features (such as a number following a mathematical symbol, i.e., + or -) rather than understanding mathematical principles. When students commit to shallow learning, they might be able to solve problems with a

particular surface feature. However, such knowledge does not transfer to other problems with different surface appearance but should be solved with the same mathematical principle.

For example, students might learn to subtract b from $ax + b = c$ where a , b , and c are numbers. This is an application of a mathematical principle of balancing an equation by applying a same operation to both sides of the equation. If students learn this principle only with the problems where b follows a plus sign, they might subtract b from $ax - b = c$ (which indeed is one of the most frequently observed common errors that students make (Booth and Koedinger 2008; Matsuda et al. 2009)). For Learning by Teaching, we observed that when SimStudent committed to shallow learning, the student's learning also tended to be shallow (Matsuda et al. 2012). In this past study, the shallow learning happens most notably on equations with variables on both sides, presumably because there are many combinations of the element-level features (e.g., the order of variable and constant terms, a sign of a term, etc.).

The above-mentioned findings on the suspicious shallow learning motivated us to structure the quiz with the Final Challenge, which is indeed the same difficulty level as Equations with Variables on both Sides (i.e., Final Challenge contains 8 more equations with variables on both sides). This means that, to pass the quiz, students must solve a set of 4 equations with variables on both sides and then another 8 equations of the same type. Since these 12 equations are carefully designed not to have the same surface features (e.g., the order of variable and constant terms, the signs of terms, etc.), it is therefore unlikely that students can pass the quiz merely learning surface features. Furthermore, the Equation section of the online test have 10 equations that never appear in the quiz. Therefore, it is arguably unlikely that our measure failed to detect students' "shallow" learning. The shallowness of the "shallow" learning is, of course, open to question. Further investigation on the skill transfer (e.g., near vs. far transfer) is necessary.

It is surprising that the current data show that Goal-Oriented Practice (APLUSTUTOR), which is driven by a cognitive tutor without global student modeling (i.e., knowledge tracing), is as effective as a fully functional cognitive tutor (COGTUTOR+) with adaptive problem selection. Goal-Oriented Practice is a variant of cognitive tutoring that does not have knowledge tracing (it is only equipped with model tracing to provide immediate feedback and just-in-time hint). Instead, APLUSTUTOR allows students to enter problems to practice by themselves while it also provides a pre-compiled set of quiz problems and Problem Bank. It is therefore an interesting version of the double-loop model for an intelligent tutor (VanLehn 2006) with an outer-loop that is controlled by the students selecting problems by themselves.

Studies show that an adaptive problem selection made by an intelligent tutoring system yields better learning than random selection (Metcalfé and Kornell 2005) and fixed order (Corbett 2000). The current study showed that cognitive tutoring on problems that students selected by themselves along with a fixed set of problems (i.e., Problem Bank) is as effective as cognitive tutoring with adaptive problem selection. However, the current findings have an obvious confounding with the presence of quiz that allowed students to obey leaning by editing. Since APLUSTUTOR is *goal oriented*—students must pass a set of pre-defined quiz by themselves, students were simply able to "adaptively" enter a failed quiz item as a practice problem on the cognitive tutor, and have the tutor provide scaffolding on how to solve it. Further

investigation on the effect of problem selection (adaptive vs. self-selection vs. goal oriented, etc.) in the context of learning by editing is necessary to advance the theory of adaptive tutoring.

An extension of this line of research on the problem selection is to have APLUS provide a student with the next problem to teach based on the student's competency computed by model tracing. For example, the teachable agent could ask students to teach a particular problem next. The student's competency can be computed based on the accuracy of feedback and hints that the student provided to the teachable agent. This might be interesting for future research.

The current two studies show relatively small effect sizes. Although a meta-analysis reported that learning by teaching in face-to-face settings tended to show a small effect size (Roscoe and Chi 2007), we anticipate that the effect of learning by teaching will be amplified with the adaptive technology support. Further investigation is necessary to understand if and how innovative learning technology can magnify the effect of tutor learning.

Conclusion

We found that learning by teaching a teachable agent with metacognitive scaffolding on how to teach is effective for students with various levels of prior competency, and it is as effective as learning by being tutored across all levels of students' prior competency. The lessons learned on the importance of metacognitive scaffolding from the current study provide insights into a successful implementation of a teachable agent that promotes the tutor-learning effect.

In the current article, two versions of cognitive tutors were implemented—one with a traditional mastery learning with an adaptive problem selection based on students' competency (Cognitive Tutoring) and another one that does not provide mastery learning but students needed to pass the quiz by themselves while receiving cognitive tutoring and metacognitive scaffolding (Goal-Oriented Practice).

We also compared learning processes for Learning by Teaching, Goal-Oriented Practice, and Cognitive Tutoring. There was no notable difference in the way students received metacognitive scaffolding between LBT and GOP. For both conditions, students rarely request metacognitive hints by themselves, but instead mostly received those that Mr. Williams, the meta-tutor agent, proactively provided.

Learning by teaching is a promising style of learning with a proven effect in the current literature. Developing an effective online environment with a teachable agent will therefore make a significant contribution to students' learning with a substantial impact on the current education system. Although our current implementation of learning by teaching (APLUS) produces an actual learning gain (measured by pre- and post-tests), its effect size is relatively small. The current work is focused on a value added by the metacognitive scaffolding to learning by teaching. Additional research is needed to further enhance the effect of tutor learning—e.g., letting the teachable agent ask constructive and reflective questions might be an interesting research topic.

Acknowledgements The research reported here was partly supported by the National Science Foundation through Grant No. 1643185 to Texas A&M University, and U.S. Department of Education (the Institute of Education Sciences) through Grant R305A180319 to North Carolina State University. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education. The author would also like to thank to Qing Li and Wenxin Zhang at the Innovative Educational Computing Laboratory who contributed to data analysis.

References

- Anderson, J. R., Boyle, C. F., Corbett, A. T., & Lewis, M. W. (1990). Cognitive modeling and intelligent tutoring. *Artificial Intelligence*, 42(1), 7–49.
- Anderson, J. R., Boyle, C. F., & Reiser, B. J. (1985). Intelligent tutoring systems. *Science*, 228(4698), 456–462.
- Anderson, J. R., Corbett, A. T., Fincham, J. M., Hoffman, D., & Pelletier, R. (1992). General principles for an intelligent tutoring architecture. Regian, J. Wesley, Shute, Valerie J. (Eds). *Cognitive approaches to automated instruction* (pp. 81–103). Hillsdale: Lawrence Erlbaum Associates, Inc. viii, 255 pp. *SEE BOOK*.
- Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *Journal of the Learning Sciences*, 4(2), 167–207.
- Anderson, J. R., & Pelletier, R. (1991). A development system for model-tracing tutors. *Proc. of the International Conference on the Learning Sciences*, 1–8.
- Annis, L. F. (1983). The processes and effects of peer tutoring. *Human Learning*, 2(1), 39–47.
- Baker, R., Corbett, A. T., Gowda, S. M., Wagner, A. Z., MacLaren, B. A., Kauffman, L. R., et al. (2010). Contextual Slip and Prediction of Student Performance after Use of an Intelligent Tutor. In P. De Bra, A. Kobsa, & D. Chin (Eds.), *User Modeling, Adaptation, and Personalization: 18th International Conference, UMAP 2010, Big Island, HI, USA, June 20–24, 2010. Proceedings* (pp. 52–63). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Baker, R., Corbett, A. T., Roll, I., & Koedinger, K. R. (2008). Developing a generalizable detector of when students game the system. *User Modeling and User-Adapted Interaction*, 18(3), 287–314.
- Biswas, G., Jeong, H., Kinnebrew, J. S., Sulcer, B., & Roscoe, R. D. (2010). Measuring self-regulated learning skills through social interactions in a teachable agent environment. *Research and Practice in Technology Enhanced Learning*, 123–152.
- Biswas, G., Leelawong, K., Schwartz, D., Vye, N., & Vanderbilt, T. T. A. G. A. (2005). Learning by teaching: a new agent paradigm for educational software. *Journal Applied Artificial Intelligence*, 19(3&4), 363–392.
- Biswas, G., Segedy, J. R., & Bunchongchit, K. (2016). From design to implementation to practice a learning by teaching system: Betty’s brain. *International Journal of Artificial Intelligence in Education*, 26(1), 350–364. <https://doi.org/10.1007/s40593-015-0057-9>.
- Booth, J. L., & Koedinger, K. R. (2008). Key misconceptions in algebraic problem solving. In B. C. Love, K. McRae, & V. M. Sloutsky (Eds.), *Proceedings of the 30th annual conference of the cognitive science society* (pp. 571–576). Austin: Cognitive Science Society.
- Bredeweg, B., Gómez-Pérez, A., André, E., & Salles, P. (2009). DynaLearn - engaging and informed tools for learning conceptual system knowledge. In P. R. R. Azevedo & G. Biswas (Eds.), *Cognitive and metacognitive educational systems (MCES2009)*, AAAI fall symposium (pp. 46–51). Arlington: AAAI Press.
- Bredeweg, B., Liem, J., Beek, W., Linnebank, F., Gracia, J., Lozano, E., et al. (2013). DynaLearn – An intelligent learning environment for learning conceptual knowledge. *AI Magazine*, 34(4), 46–65.
- Britz, M., Dixon, J., & McLaughlin, T. (1989). The effects of peer tutoring on mathe- matics performance: A recent review. *B. C. Journal of Special Education*, 13(1), 17–33.
- Chase, C., Chin, D., Oppezzo, M., & Schwartz, D. (2009). Teachable agents and the Protégé effect: Increasing the effort towards learning. *Journal of Science Education and Technology*, 18(4), 334–352.
- Cohen, E. G. (1994). Restructuring the classroom: Conditions for productive small groups. *Review of Educational Research*, 64(1), 1–35. <https://doi.org/10.2307/1170744>.
- Cohen, P. A., Kulik, J. A., & Kulik, C. L. C. (1982). Education outcomes of tutoring: A meta-analysis of findings. *American Educational Research Journal*, 19(2), 237–248.

- Cohen, W. W. (1998). Hardness results for learning first-order representations and programming by demonstration. *Machine Learning*, 30(1), 57–87.
- Cook, S. B., Scruggs, T., Mastropieri, M., & Casto, G. (1986). Handicapped students as tutors. *Journal of Special Education*, 19(4), 483–492.
- Corbett, A. T. (2000). Cognitive Maturity learning in the ACT programming tutor: AAAI technical report SS-00-01.
- Corbett, A. T., & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4), 253–278.
- Devin-Sheehan, L., Feldman, R. S., & Allen, V. L. (1976). Research on children tutoring children: A critical review. *Review of Educational Research*, 46(3), 355–385. <https://doi.org/10.2307/1170008>.
- Fuchs, L. S., Fuchs, D., Hamlett, C. L., Phillips, N. B., Karns, K., & Dutka, S. (1997). Enhancing Students' helping behavior during peer-mediated instruction with conceptual mathematical explanations. *Elementary School Journal*, 97(3), 223–249.
- Gartner, A., Kohler, M., & Riessman, F. (1971). *Children teach children: Learning by teaching*. New York, NY: Harper & Row.
- Hedin, D. (1987). Students as teachers: A tool for improving school. *Social Policy*, 17(3), 42–47.
- Jun, Y. (2003). Facilitating mathematical learning with a peer tutoring system: Lessons learned. *Journal of Computers in Mathematics and Science Teaching*, 22(1), 75–92.
- King, A., Staffieri, A., & Adelgais, A. (1998). Mutual peer tutoring: Effects of structuring tutorial interaction to scaffold peer learning. *Journal of Educational Psychology*, 90(1), 134–152. <https://doi.org/10.1037/0022-0663.90.1.134>.
- Klingner, J., & Vaughn, S. (1996). Reciprocal teaching of reading comprehension strategies for students with learning disabilities who use English as a second language. *Elementary School Journal*, 96(3), 275–293.
- Lau, T. A., & Weld, D. S. (1998). *Programming by demonstration: An inductive learning formulation Proceedings of the 4th international conference on Intelligent user interfaces* (pp. 145–152). New York: ACM Press.
- Lenat, D., & Durlach, P. (2014). Reinforcing math knowledge by immersing students in a simulated learning-by-teaching experience. *International Journal of Artificial Intelligence in Education*, 1–35. <https://doi.org/10.1007/s40593-014-0016-x>.
- Matsuda, N., Barbalios, N., Zhao, J., Ramamurthy, A., Stylianides, G., & Koedinger, K. R. (2016). Tell me how to teach, I'll learn how to solve problems. In A. Micarelli, J. Stamper, & K. Panourgia (Eds.), *Proceedings of the international conference on intelligent tutoring systems* (pp. 111–121). Switzerland: Springer.
- Matsuda, N., Cohen, W. W., & Koedinger, K. R. (2005). Applying programming by demonstration in an intelligent authoring tool for cognitive tutors AAAI *Workshop on Human Comprehensible Machine Learning (Technical Report WS-05-04)* (pp. 1–8). Menlo Park, CA: AAAI association.
- Matsuda, N., Cohen, W. W., & Koedinger, K. R. (2015). Teaching the teacher: Tutoring SimStudent leads to more effective cognitive tutor authoring. *International Journal of Artificial Intelligence in Education*, 25, 1–34. <https://doi.org/10.1007/s40593-014-0020-1>.
- Matsuda, N., Griger, C. L., Barbalios, N., Stylianides, G., Cohen, W. W., & Koedinger, K. R. (2014). Investigating the effect of meta-cognitive scaffolding for learning by teaching. In S. Trausen-Matu, K. Boyer, M. Crosby, & K. Panourgia (Eds.), *Proceedings of the international conference on intelligent tutoring systems* (pp. 104–113). Switzerland: Springer.
- Matsuda, N., Keiser, V., Raizada, R., Tu, A., Stylianides, G. J., Cohen, W. W., & Koedinger, K. R. (2010). Learning by teaching SimStudent: Technical accomplishments and an initial use with students. In V. Aleven, J. Kay, & J. Mostow (Eds.), *Proceedings of the international conference on intelligent tutoring systems* (pp. 317–326). Heidelberg, Berlin: Springer.
- Matsuda, N., Lee, A., Cohen, W. W., & Koedinger, K. R. (2009). A computational model of how learner errors Arise from weak prior knowledge. In N. Taatgen & H. van Rijn (Eds.), *Proceedings of the annual conference of the cognitive science society* (pp. 1288–1293). Austin: Cognitive Science Society.
- Matsuda, N., Yarzebinski, E., Keiser, V., Raizada, R., Cohen, W. W., Stylianides, G. J., & Koedinger, K. R. (2012). Shallow learning as a pathway for successful learning both for tutors and tutees. In N. Miyake, D. Peebles, & R. P. Cooper (Eds.), *Proceedings of the 34th annual conference of the cognitive science society* (pp. 731–736). Austin: Cognitive Science Society.
- Matsuda, N., Yarzebinski, E., Keiser, V., Raizada, R., Stylianides, G. J., Cohen, W. W., & Koedinger, K. R. (2011). Learning by teaching SimStudent – An initial classroom baseline study comparing with cognitive tutor. In G. Biswas & S. Bull (Eds.), *Proceedings of the international conference on artificial intelligence in education* (pp. 213–221). Berlin, Heidelberg: Springer.

- Matsuda, N., Yarzebinski, E., Keiser, V., Raizada, R., Stylianides, G. J., & Koedinger, K. R. (2013a). Studying the effect of a competitive game show in a learning by teaching environment. *International Journal of Artificial Intelligence in Education*, 23(1-4), 1–21 [invited for the special issue on the best of ITS2012].
- Matsuda, N., Yarzebinski, E., Keiser, V., Raizada, R., William, W. C., Stylianides, G. J., & Koedinger, K. R. (2013b). Cognitive anatomy of tutor learning: Lessons learned with SimStudent. *Journal of Educational Psychology*, 105(4), 1152–1163. <https://doi.org/10.1037/a0031955>.
- Metcalfe, J., & Kornell, N. (2005). A region of proximal learning model of study time allocation. *Journal of Memory and Language*, 52(4), 463–477. <https://doi.org/10.1016/j.jml.2004.12.001>.
- Michie, D., Paterson, A., & Hayes, J. E. (1989). *Learning by teaching Proc. of Second Scandinavian Conference on Artificial Intelligence* (pp. 413–436). Amsterdam: IOS.
- Mioduser, D., Zuzovsky, R., Salles, P., Souza, A., Portella, A., Braga, E., . . . Stelzhammer, M. (2012). Final report on DynaLearn evaluation studies (EC FP7 STREP project 231526, deliverable D7.4): DynaLearn.
- Mitchell, T. M. (1982). Generalization as search. *Artificial Intelligence*, 18(2), 203–226.
- Muggleton, S., & de Raedt, L. (1994). Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19-20(supplement 1), 629–679.
- Palincsar, A. S., & Brown, A. L. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition and Instruction*, 1(2), 117–175.
- Pane, J. F., Griffin, B. A., McCaffrey, D. F., & Karam, R. (2014). Effectiveness of cognitive tutor algebra I at scale. *Educational Evaluation and Policy Analysis*, 36(2), 127–144. <https://doi.org/10.3102/0162373713507480>.
- Pareto, L. (2014). A teachable agent game engaging primary school children to learn arithmetic concepts and reasoning. *International Journal of Artificial Intelligence in Education*, 24(3), 251–283. <https://doi.org/10.1007/s40593-014-0018-8>.
- Renkl, A. (1995). Learning for later teaching: An exploration of mediational links between teaching expectancy and learning results. *Learning and Instruction*, 5(1), 21–36. [https://doi.org/10.1016/0959-4752\(94\)00015-h](https://doi.org/10.1016/0959-4752(94)00015-h).
- Ritter, S., Anderson, J. R., Koedinger, K. R., & Corbett, A. (2007). Cognitive tutor: Applied research in mathematics education. *Psychonomic Bulletin & Review*, 14(2), 249–255.
- Robinson, D., Schofield, J., & Steers-Wentzell, K. (2005). Peer and cross-age tutoring in math: Outcomes and their design implications. *Educational Psychology Review*, 17(4), 327–362.
- Rohrbeck, C. A., Ginsburg-Block, M., Fantuzzo, J. W., & Miller, T. R. (2003). Peer-assisted learning interventions with elementary school students: A meta-analytic review. *Journal of Educational Psychology*, 95(2), 240–257. <https://doi.org/10.1037/0022-0663.95.2.240>.
- Roscoe, R. D., & Chi, M. T. H. (2007). Understanding tutor learning: Knowledge-building and knowledge-telling in peer tutors' explanations and questions. *Review of Educational Research*, 77(4), 534–574.
- Roscoe, R. D., Segedy, J. R., Sulcer, B., Jeong, H., & Biswas, G. (2013). Shallow strategy development in a teachable agent environment designed to support self-regulated learning. *Computers & Education*, 62, 286–297. <https://doi.org/10.1016/j.compedu.2012.11.008>.
- Russell, S. J., & Norvig, P. (2003). *Artificial intelligence: A modern approach* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.
- Schwartz, D., Blair, K. P., Biswas, G., & Leelawong, K. (2007). Animations of thought: Interactivity in the teachable agent paradigm. In R. Lowe & W. Schnotz (Eds.), *Learning with animation: Research and implications for design* (pp. 114–140). Cambridge: Cambridge University Press.
- Sharpley, A. M., Irvine, J. W., & Sharpley, C. F. (1983). An examination of the effectiveness of a cross-age tutoring program in mathematics for elementary school children. *American Educational Research Journal*, 20(1), 103–111.
- Tan, J., Biswas, G., & Schwartz, D. (2006). Feedback for metacognitive support in learning by teaching environments. In R. Sun & N. Miyake (Eds.), *Proceedings of the 28th annual meeting of the cognitive science society* (pp. 828–833). Vancouver.
- Uresti, J. A. R., & du Boulay, B. (2004). Expertise, motivation and teaching in learning companion systems. *International Journal of Artificial Intelligence in Education*, 14(2), 193–231.
- VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education*, 16.
- Walker, E., Rummel, N., & Koedinger, K. R. (2014). Adaptive intelligent support to improve peer tutoring in algebra. *International Journal of Artificial Intelligence in Education*, 24(1), 33–61. <https://doi.org/10.1007/s40593-013-0001-9>.
- Zhao, G., Ailiya, & Shen, Z. (2012). Learning-by-teaching: Designing teachable agents with instructional motivation. *Educational Technology & Society*, 15, 62–74.