Testing the Split Attention Effect on Learning in a Natural Educational Setting Using an Intelligent Tutoring System for Geometry

Robert GM Hausmann (bhausmann@carnegielearning.com)
Annalies Vuong (avuong@carnegielearning.com)
Carnegie Learning, Inc.
437 Grant Street, Frick Building, 20th Floor
Pittsburgh, PA 15219 USA

Abstract

Intelligent tutoring systems (ITS) are a successful application of cognitive science theory to the field of education. Data generated by students using an ITS can also be used to test the external validity of cognitive science principles developed largely in laboratory settings. The present paper collected data from high-school students using two versions of *Cognitive Tutor*, an ITS for Geometry, to assess the impact of eliminating the split-attention effect. The two versions differed in the extent to which the interface required split attention during problem solving. One version used integrated diagrams whereas the other used non-integrated tables and diagrams. Results suggested that students needed fewer problems to master skills in the integrated version, and this was particularly true for mastering difficult skills. This study demonstrates the successful use of cognitive science principles to improve learning through empirically and theoretically derived enhancements to an ITS used in a natural educational setting.

Keywords: Intelligent tutoring systems, mathematics instruction, split-attention effect, cognitive load theory.

Introduction

One of the promises of cognitive science is that it informs the design and implementation of effective instruction and educational tasks (Bruer, 1997). Unfortunately, there is often a disconnect between instructional tasks, as they are originally designed, and the actual implementation of those tasks in the classroom (Stein, Smith, Henningsen, & Silver, 2000; p. 4). One way to partially mitigate this danger is to design instructional tasks in software. Ideally, the design of the software is based on a cognitive theory of learning. Several intelligent tutoring systems (ITS) have been designed based on cognitive theories, including constraint-based reasoning (Mitrovic & Ohlsson, 1999), failure-driven learning (VanLehn, 1988), and the ACT-R theory of human cognition (Anderson, Boyle, Corbett, & Lewis, 1990).

The designers of intelligent tutoring systems face at least two challenges. First, they must demonstrate a learning benefit above and beyond traditional classroom materials and activities. More importantly, an ITS should be able to demonstrate continuous improvements to learning as the theories and empirical findings the from cognitive and learning sciences advance.

The purpose of the current paper is to evaluate how enhancements to the ITS *Cognitive Tutor: Geometry* affected student learning in real-world, educational settings. The ITS modifications were based on the predictions of cognitive load theory, which claims that learning is harmed when a student splits his or her attention across interdependent sources of information. The so-called "split-attention effect" inspired an *in vivo* study in which a single unit from *Cognitive Tutor* was heavily revised to reduce split attention caused by the user interface. The goal of this paper is to extend the generalizability of that *in vivo* experiment by conducting a more in-depth analysis of student learning using data collected from real students using two different versions of the commercially available *Cognitive Tutor*.

Cognitive Tutor

Cognitive Tutor is an intelligent tutoring system inspired by the ACT-R theory of human cognition. *Cognitive Tutor* is based on the pedagogical principle that knowledge is decomposed into knowledge components called *skills*, and learning is maximized when the student is responsible for actively taking each problem-solving step. A cognitive model tracks if the student takes a step off the ideal solution path. Student modeling also allows the tutor to provide immediate feedback, as well as help, in the form of hints, at any step during problem solving. The *Cognitive Tutor* operationally defines "mastery" when the probability that a student knows a skill reaches a threshold of 95%.

Cognitive Tutor has been evaluated for its efficacy in both the laboratory and the classroom (Anderson, Corbett, Koedinger, & Pelletier, 1995; Koedinger, Anderson, Hadley, & Mark, 1997), as well in randomized field trials (Ritter, Kulikowich, Lei, McGuire, & Morgan, 2007). A majority of the aforementioned studies used traditional learning materials, such as textbooks and paper-and-pencil homework assignments, as the baseline learning condition. Summarizing over several studies, Corbett (2001) estimates the effect size of *Cognitive Tutor* to be around one standard deviation above traditional instructional materials.

Although effective, there is still room for improvement given that one-on-one human tutoring, when combined with mastery learning, produces about a two standard deviation increase in learning (Bloom, 1984). One way to improve an ITS is to go back and reanalyze the design of individual units and ask if there are any opportunities for enhancing the student's interaction with the target material. One of the pedagogical commitments of *Cognitive Tutor* is to "minimize working memory load" (Anderson et al., 1995; p.

180). Therefore, the next section discusses cognitive load theory as it applies to geometry instruction.

Cognitive Load Theory and the Split-Attention Effect

According to cognitive load theory, learning is most likely to take place when the learning environment maximizes *germane* load and minimizes *extraneous* load. Germane cognitive load is defined as "load devoted to the processing, construction and automation of schemas;" whereas extraneous load is defined as "load generated by the manner in which information is presented to learners and is under the control of instructional designers" (Chandler & Sweller, 1991).

Solving problems in geometry is most likely to include both types of load. For example, it is often the case that geometry problems are stated verbally, and they are accompanied by a diagram. The student's first task is to map the given information, stated in the problem scenario, onto the figure. For example, this would require that the student holds an angle name and its measure in working memory (e.g., $m\angle ABC = 15°$) while locating the relevant angle in the diagram. This is an example of the *split-attention effect* (Kalyuga, Chandler, & Sweller, 1999).

Holding the angle name and its measure in working memory is not directly relevant to learning how to solve these types of problems; therefore, the working memory load imposed on the student is considered an extraneous load. According to cognitive load theory, instructional designers are recommended that they create a learning environment that minimizes extraneous load caused by the split-attention effect.

Based on the hypothesis that splitting one's attention across multiple sources of information harms learning, Butcher and Aleven (2008) conducted an *in vivo* experiment where they contrasted classroom learning from two different versions of *Cognitive Tutor: Geometry*. The traditional interface included a verbal statement of the given information, a diagram, and a table. The table was the focus of the student interactions and required students to enter the measure of each angle, as well as a reason justifying the calculation. For the experimental interface, all of the inputs were made directly in the diagram. Students entered their measures and reasons by clicking on angles in the diagram. The learning results from that study suggested that the interactive diagram was easier to learn from, especially in terms of a delayed posttest for numerical test items.

Because the study by Butcher and Aleven (2008) was conducted with a relatively restricted sample of students (n = 58) and a single unit of instruction, there is an open question as to whether a change to the interface translates to classroom learning. Will the results replicate when they are implemented in the "wild?"

To address this question, we conducted an analysis of log files generated by students using one of two different versions of *Cognitive Tutor: Geometry* that differed in terms of the split attention required by the user interface.

Method

Participants

We compared the usage data from two different versions of *Cognitive Tutor: Geometry* software developed by Carnegie Learning, which are described in the section below. User log files generated by the tutor contain detailed information about every action taken in the interface, including latencies, errors, and access to the various forms of help (i.e., requesting hints, accessing the glossary, reading the lesson page, or studying the interactive example).

Two cohorts of students used two different versions of the software. Approximately 10% of the schools that use Carnegie Learning products were randomly selected to collect log files from their students. For the current study, that translates into approximately n = 1,577 students for the 2009 version and n = 2,168 students for the 2010 version.

Materials

The interface for several units and sections of the *Cognitive Tutor: Geometry* curriculum were revised to reduce the split-attention effect by using interactive diagrams. Those units include: Pythagorean Theorem, Angle Relationships in a Triangle, and Special Right Triangles.

Table Interface (v.2009). Previous versions of *Cognitive Tutor: Geometry* included an interface similar to the one described as the control condition from Butcher and Aleven (2008). The interface included a static diagram, a verbal statement (in paragraph form) of the givens and the sought, and a table of angles in which the student is tasked with calculating the measure and providing a rationale for the calculation (see Fig. 1).

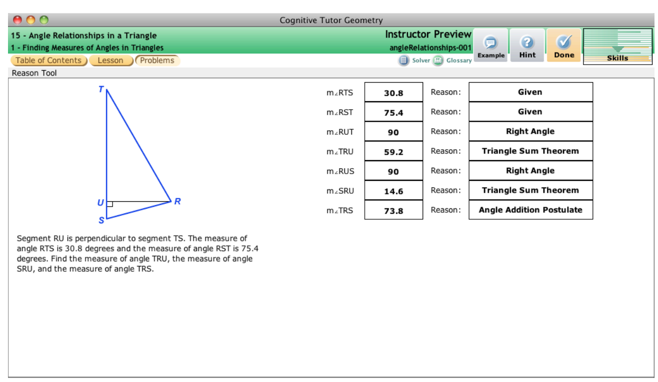

Figure 1: Table Interface (v.2009).

Interactive Diagram Interface (v.2010). In an effort to reduce the split-attention effect, several units/sections of geometry were modified to use an "interactive diagram." The design of the interactive diagrams was similar, but not identical, to the design of the circle tutor used in Butcher and Aleven (2008). All student interactions were handled in the diagram itself. Students had the ability to click on individual angles. Once an angle was selected, a *flyout* (shown with a blue background in Fig. 2) appeared with several input fields, including the angle measure, a reason field where the student justifies her calculation, and drop-down menus to select the other angles that participate in the target angle's calculation. When the tutor determined that each entry was complete, a summary appeared under the "Diagram Notes," and the diagram itself was labeled with the angle measure.

As previously mentioned, the revised design was intended to reduce working memory load by externalizing some of the information. Because students can *see* an angle in the diagram, they are no longer burdened with holding the angle's name and measure in working memory. Theoretically, this should provide students with more cognitive resources to search the problem space (Larkin & Simon, 1987) and generate domain-relevant inferences.

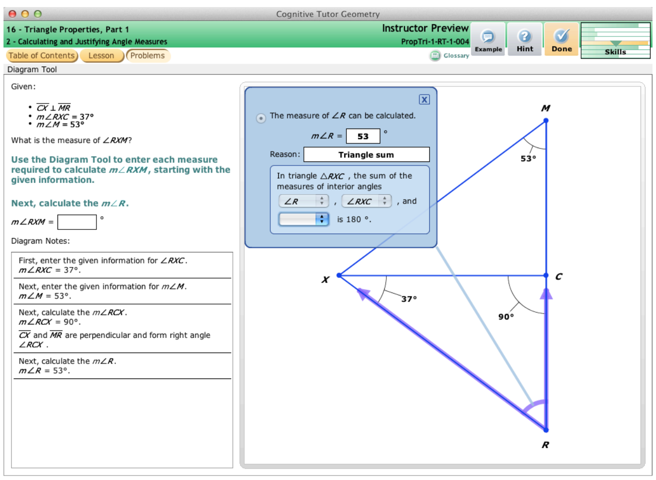

Figure 2: Interactive Diagram Interface (v.2010).

Results

Due to the large sample sizes, all of the independent, two-sample tests were significant with an alpha level of $\alpha = .01$; thus, instead of reporting *p*-values, we relied on Cohen's *d* as an effect-size indicator. This allows for a better estimate of the practical significance of the differences.

The results are broken down into two sections. The first section analyzes learning at the unit level. Given that units from v.2009 were reorganized in v.2010, this made one-to-one comparisons at the unit level difficult. We therefore analyzed learning at a finer level of granularity by focusing on the learning of individual skills, regardless of the unit in which the skill appeared.

To control for school-related differences, we replicated all skill-level analyses by restricting our sample to students from the same school. There were small numeric differences in the magnitude of the effect sizes, but they were all in the same direction and interpretation category (e.g., small [d = .20], medium [d = .50], & large [d = .80]); therefore, we collapsed across schools in all subsequent analyses.

Learning Measured at the Unit Level

The software organizes the learning material hierarchically. Units form the highest level of organization, which are sub-divided into sections. Sections are further broken down into problems, which can be further subdivided into individual skills.

We used two different measures to evaluate learning at the unit level. The first was the median number of *Problems* the students solved before graduating the unit. Graduation was defined as mastering all skills. Second, we measured the total amount of *Time* (in minutes) spent in the unit. The results for unit-level measures are summarized in Table 1.

For the problem metric, students using the table interface generally needed to solve more problems in the tutoring system than students using the interactive diagram interface. This was true for Special Right Triangles (5+33 vs. 27 in v.2009 and v.2010, respectively) and Angle Relationships (39 vs. 9+7). The results were reversed, however, for the Pythagorean Theorem unit (5 vs. 9).

The results for the total amount of time spent in the tutor were largely consistent (and correlated with) the number of problems the students solved. The biggest time saving was observed for Angle Relationships (220.01 vs. 70.67+32.22).

Table 1: Unit-level comparisons between the two versions of the software.

Unit	Year	Section Num.	*n*	Problems to Grad.	Time (min.)
Pythagorean	2009	1	1,577	5	19.34
Theorem	2010	1	2,168	9	38.54
Special	2009	1	858	5	9.67
Right		2	745	33	54.56
Triangles	2010	1	1,197	27	67.71
Angle	2009	1	331	39	220.01
Relationships	2010	1	1,213	9	70.67
		2	899	7	32.22

One potential explanation for the results in which the table interface demonstrated better performance than the interactive diagram interface could be due to a difference in the distribution of material across sections. For example, the Special Right Triangles unit originally included two sections: "Finding the Lengths of Sides of a 45-45-90 Triangle" and "Finding the Lengths of Sides of a 30-60-90 Triangle." In the revised version, these sections were combined to form a single section: "Calculating the Lengths of Sides of Special Right Triangles." Combining the sections may have increased the difficulty because students were required to discriminate between the principles necessary to solve two different types of problems.

A similar case could be made for the Pythagorean Theorem unit. The original unit included a section that only required students to solve for the length of the hypotenuse. However, in the revised version, either the hypotenuse or the leg could be the sought value. Again, students were required to make finer-grained discriminations in the revised version, which could have accounted for the increased time required to graduate from the unit.

Learning Measured at the Skill Level

As the previous paragraph suggests, the two versions of *Cognitive Tutor* changed in more ways than just the interface. In some cases, the geometry units were rearranged such that the section breakdown was different from one year to the next; therefore, measures of learning could be confounded by section-level changes.

To control for these potentially confounding factors, we measured learning on individual skills, with learning operationally defined as the number of problems the students solved to master each skill. As stated previously, mastery was achieved when the probability of a student knowing a skill reached 95%. To ensure a fair comparison, for this analysis we focused on skills that were consistent between the two versions.

The skills for each unit are presented in separate sections below. The name and id of each skill can be found in the Appendix.

Special Right Triangles. Two types of triangles were covered in this section: 45-45-90 and 30-60-90. For the easier triangles (45-45-90), the revised version using interactive diagrams actually led to worse performance in that students needed more problems to master these skills in the revised interface (see Table 2; shaded values). Effect sizes ranged from small (d = -0.33) to large (d = -2.54).

The reverse, however, was true for the more challenging triangles (30-60-90). The revised interface reduced the number of problems needed to master the associated skills. Effect sizes ranged between (d = .33 - .46).

Table 2: Skill comparisons for Special Right Triangles.

	2009		2010		
Skill ID	***n***	**$\bar{x}$ (*SD*)**	***n***	**$\bar{x}$ (*SD*)**	***d***
SR-45_01	871	3.09 (1.9)	991	3.99 (3.41)	-0.33
SR-45_02	785	1.52 (1.06)	1028	6.32 (7.66)	-0.88
SR-45_03	819	3.33 (1.2)	938	16.67 (7.81)	-2.39
SR-45_04	798	2.35 (1.03)	934	16.6 (7.88)	-2.54
SR-30_01	476	21.78 (11.36)	893	17.27 (7.79)	0.46
SR-30_02	558	19.68 (11.8)	991	16.29 (7.65)	0.34
SR-30_03	478	21.56 (12.07)	1000	16.63 (6.96)	0.50
SR-30_04	543	21.08 (12.13)	1009	17.89 (5.93)	0.33
SR-30_05	577	16.29 (10.95)	924	19.5 (5.85)	-0.36
SR-30_06	465	22.31 (12.04)	890	17.57 (7.95)	0.46

Note: Skill IDs refer to **S**pecial **R**ight Triangles, followed by the fist angle measure (e.g., 45 or 30).

One potential explanation for the inconsistent results is that the previous version was easier than the revised version because it separated the two special right triangles into their own sections (see Table 1). When students solved 45-45-90 problems, they did not have to discriminate between shorter, longer, or equal leg lengths when calculating the non-given side. Restricting our analyses to the 2009 sample, students demonstrated fewer errors while solving 45-45-90 problems (M = 5.15, SD = 5.93) than 30-60-90 problems (M = 39.53, SD = 37.35), d = 1.29. This suggests that, at the section level, the 45-45-90 problems were easier to solve.

Angle Relationships in a Triangle. The skills associated with the unit "Angle Relationships in a Triangle" were more consistent. For these skills, the revised interface showed a marked reduction in the average number of problems the students solved before mastering their skills. The effect sizes ranged between medium (d = .57) and large (d = 2.66), with a majority of the skills falling in the large category (see Table 3).

The lone exception was the first skill, which asks the students to "Enter given value." It seems that entering the given value was slightly easier in the original table interface that required students to map between the verbal description and entering the given in a table. This might be because the answer of the top row of the table is *always* the given value, whereas a small amount of search is required to enter the given in the interactive diagram.

Table 3: Skill comparisons for Angle Relationships.

	2009		2010		
Skill ID	***n***	**$\bar{x}$ (*SD*)**	***n***	**$\bar{x}$ (*SD*)**	***d***
Ang_Re_01	1140	1.86 (1.27)	901	2.29 (0.98)	-0.38
Ang_Re_02	1244	4.38 (5.83)	891	1.79 (1.56)	0.61
Ang_Re_03[1]	455	9.86 (5.43)	887	2.19 (2.49)	1.82
Ang_Re_04	455	5.90 (8.8)	887	2.19 (2.49)	0.57
Ang_Re_05	369	24.50 (16.83)	887	2.19 (2.49)	1.85
Ang_Re_06	344	29.80 (16.41)	887	2.19 (2.49)	2.35
Ang_Re_07	344	30.17 (16.35)	887	2.19 (2.49)	2.39
Ang_Re_08	331	31.86 (15.59)	887	2.19 (2.49)	2.66
Ang_Re_09	344	29.79 (16)	887	2.19 (2.49)	2.41
Ang_Re_10	450	10.10 (13.31)	889	3.88 (2.2)	0.65

Pythagorean Theorem. The final section in which there were matching skills was the unit on the Pythagorean Theorem. The two skills in this section that fit our criteria both demonstrated an advantage for the interactive diagram. Students using the revised interface needed fewer problems to solve both skills (i.e., calculate the length of the

[1] "Ang_Re_03" through "_09" have the same statistics because the 2010 skill included each of the 2009 variants as sub-skills.

hypotenuse in and out of a contextual scenario). The effect sizes were both considered "large" ($d > .90$; see Table 4).

Table 4: Skill comparisons for Pythagorean Theorem.

	2009		2010		
Skill ID	***n***	$\bar{x}$ **(*SD*)**	***n***	$\bar{x}$ **(*SD*)**	***d***
Pythag_01	1663	15.64 (8.79)	2338	8.84 (4.42)	0.98
Pythag_02	1702	15.38 (8.9)	2338	8.84 (4.42)	0.93

Discussion

Using data gathered from the use of an intelligent tutoring system (ITS) in natural educational settings, the current study demonstrates how an already effective intelligent tutoring system can be further refined through the application of cognitive and learning theories. The current study draws from research on the "split-attention effect," which demonstrates that performance on a task is greatly reduced when the student must split her attention across interdependent sources of information. Learning is greatly reduced because working memory is tasked with holding a large number of chunks of information. According to cognitive load theory, when that large burden on working memory is not relevant to abstracting principles from the domain, then this leads to an "extraneous load." Students are not able to transfer the knowledge that is inferred from problem solving to long-term memory.

The split-attention effect is particularly relevant to solving geometry problems in an ITS, where the student is required to split her attention across a verbal scenario that states given information, a diagram that depicts relationships between segments and angles, and a table that holds information about each angle.

Butcher and Aleven (2008) demonstrated, in an *in vivo* study, that a revised ITS interface can enhance learning both immediately and over the long term. On the basis of their strong results, the *Cognitive Tutor: Geometry* interface was revised to emulate the same type of interaction. With the "interactive diagrams," students were given the chance to concentrate the focus of their attention on the learning materials.

Although the *in vivo* results were strong, there was a chance that crucial design elements did not get directly translated into the commercial version of the software. A comparison of screenshots between the Butcher and Aleven (2008; Fig. 1) and the current study (Figs. 1 & 2) reveals that there were subtle design differences. For example, the original study modified a unit on circles, whereas the current study mainly concentrated on triangles. There may be subtle content differences that lend themselves more or less well to learning gains through interactive diagrams. Second, the information in the interactive circle diagrams was echoed in a table; whereas, the triangle tutor included a "Diagram Notes" panel with similar, but differently formatted, information.

The current results support the generalization that small design differences can have a measurable impact on learning. At the unit level, there were generally mixed results with some, but not all, units demonstrating a reduced amount of time spend solving problems. Analysis at the more fine-grained level of individual skills yielded more consistent results. Most, but not all, of the skills associated with the interactive diagram showed a positive effect. Some of the skills that showed an increase in the number of problems required to master the skills seemed to fit into one of two categories. Either the skills were very easy (i.e., "enter given") or they were embedded in a particularly easy unit. In these cases, it might have been better to rely on the old design. For more difficult skills, however, there was a definite advantage to interacting directly with the diagram.

Although the results are encouraging, the current set of analyses could be improved in the following ways. First, this was not an experimental study. Students were not randomly assigned to condition; therefore, the conclusions that we can draw from these analyses are strictly correlational. However, these results are suggestive and point to interesting new research projects. For example, subsequent research should test the hypothesis that interactive diagrams are especially helpful for more difficult topics.

Another improvement on the current analyses would be to assess "robust" learning, which is defined as learning that is retained over a long interval, transfers to new situations, and helps accelerate learning of subsequent material (Koedinger, Corbett, & Perfetti, 2010). Because this was an analysis of the log files generated by student users, we were not privy to the students' pre- and post-test scores. Future analyses will look at post-requisite materials available in the tutor and evaluate if there is any evidence of transfer or accelerated future learning. Although it may have taken students more problems to master the skills presented in the Special Right Triangles unit, students might be able to transfer their knowledge more accurately when they were required to struggle with deciding which rule applies within the collapsed Special Right Triangle section (e.g., desirable difficulties; Bjork, 1994).

In addition, we would also like to conduct further analyses to determine whether the changes in the design features affected the learning curves of the matched (i.e., comparable) skills.

In conclusion, it is widely acknowledged that learning geometry is challenging. As instructional designers and members of the cognitive science community, it is incumbent upon us to ensure that learning difficult science, technology, engineering, or math (STEM) topics is both efficient and robust. One way to continuously improve our methods of instruction is to keep going back and testing our learning environments against the most recent empirical and theoretical developments. The current study takes an important step in that direction.

Acknowledgments

The authors would like to thank the schools that allowed us to collect log files from their students. More importantly, we wish to thank all of the anonymous students who used our software to solve their geometry homework problems. Special thanks to Leslie Hausmann for commenting on a previous version of this paper.

References

Anderson, J. R., Boyle, C. F., Corbett, A. T., & Lewis, M. W. (1990). Cognitive modeling and intelligent tutoring. *Artificial Intelligence*, *42*, 7-49.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, *4*, 167-207.

Bjork, R. A. (1994). Memory and metamemory considerations in the training of human beings. (J. Metcalfe & A. P. Shimamura, Eds.)*Metacognition Knowing about knowing*. The MIT Press. Retrieved from http://psycnet.apa.org/psycinfo/1994-97967-009

Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, *13*(6), 4-16.

Bruer, J. T. (1997). Education and the brain: A bridge too far. *Educational Researcher*, *26*(8), 4-16.

Butcher, K. R., & Aleven, V. A. (2008). Diagram Interaction during Intelligent Tutoring in Geometry: Support for Knowledge Retention and Deep Understanding. In B. C. Love, K. McRae, & V. M. Sloutsky (Eds.), *Proceedings of the 30th Annual Conference of the Cognitive Science Society* (pp. 1736-1741). Austin, TX: Cognitive Science Society.

Chandler, P., & Sweller, J. (1991). Cognitive load theory and the format of instruction. *Cognition and Instruction*, *8*(4), 293-332.

Corbett, A T. (2001). Cognitive computer tutors: Solving the two-sigma problem. In M. Bauer, P. J. Gmytrasiewicz, & J. Vassileva (Eds.), *User Modeling* (pp. 137-147). Berlin: Springer-Verlag. doi:10.1007/3-540-44566-8

Kalyuga, S., Chandler, P., & Sweller, J. (1999). Managing split-attention and redundancy in multimedia instruction. *Applied Cognitive Psychology*, *13*(4), 351–371.

Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, *8*(1), 30-43.

Koedinger, K. R., Corbett, A. T., & Perfetti, C. (2010). *The Knowledge-Learning-Instruction (KLI) Framework: Toward Bridging the Science-Practice Chasm to Enhance Robust Student Learning*. *Cognitive Science*. Pittsburgh, PA. Retrieved from http://www.learnlab.org/documents/KLI-Framework-Tech-Report.pdf

Larkin, J. H., & Simon, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, *11*, 65-99.

Mitrovic, A., & Ohlsson, S. (1999). Evaluation of a constraint-based tutor for a database language. *International Journal of Artificial Intelligence in Education*, *10*(3-4), 238-256.

Ritter, S., Kulikowich, J., Lei, P., McGuire, C. L., & Morgan, P. (2007). What Evidence Matters? A randomized field trial of Cognitive Tutor Algebra I. In T. Hirashima, H. U. Hoppe, & S. S.-C. Young (Eds.), *Supporting Learning Flow Through Integrative Technologies* (Vol. 162, pp. 13-20). IOS Press.

Stein, M. K., Smith, M. S., Henningsen, M. A., & Silver, E. A. (2000). *Implementing Standards-Based Mathematics Instruction: A Casebook for Professional Development*. New York, NY: Teachers College Press.

VanLehn, K. (1988). Toward a theory of impasse-driven learning. In H. Mandl & A. Lesgold (Eds.), *Learning issues for intelligent tutoring systems* (pp. 19-41). New York: Springer.

Appendix

Skill ID	Skill Name
SR-45_01	Enter given side length.
SR-45_02	Calculate leg given other leg in a 45-45-90 triangle.
SR-45_03	Calculate hypotenuse in a 45-45-90 triangle.
SR-45_04	Calculate leg given hypotenuse in a 45-45-90 triangle.
SR-30_01	Calculate longer leg given shorter leg in a 30-60-90 triangle.
SR-30_02	Calculate hypotenuse given shorter leg in a 30-60-90 triangle.
SR-30_03	Calculate shorter leg given hypotenuse in a 30-60-90 triangle.
SR-30_04	Calculate hypotenuse given longer leg in a 30-60-90 triangle.
SR-30_05	Calculate shorter leg given longer leg in a 30-60-90 triangle.
SR-30_06	Calculate longer leg given hypotenuse in a 30-60-90 triangle.
Ang_Re_01	Enter given value.
Ang_Re_02	Enter calculated value.
Ang_Re_03	Enter reason of Right Angle.
Ang_Re_04	Enter reason of Triangle Sum.
Ang_Re_05	Enter reason of Angle Addition or Triangle Sum.
Ang_Re_06	Enter reason of Triangle Exterior Angle.
Ang_Re_07	Enter reason of Linear Pair.
Ang_Re_08	Enter reason of Angle Addition.
Ang_Re_09	Enter reason of Isosceles Triangle.
Ang_Re_10	Enter reason of Equilateral Triangle.
Pythag_01	Find hypotenuse in context
Pythag_02	Find hypotenuse out of context