Preprint of: VanLehn, K., Banerjee, C., Milner, F. & Wetzel, J. (2020) Teaching Algebraic Model Construction: A Tutoring System, Lessons Learned and an Evaluation. International Journal of Artificial Intelligence in Education. https://doi.org/10.1007/s40593-020-00205-3

Teaching algebraic model construction: A tutoring system, lessons learned and an evaluation

Kurt VanLehn, Chandrani Banerjee, Fabio Milner & Jon Wetzel
Arizona State University

Abstract: An algebraic model uses a set of algebra equations to precisely describe a situation. Constructing such models is a fundamental skill required by US standards for both math and science. It is usually taught with algebra word problems. However, many students still lack the skill, even after taking several algebra courses in high school and college. We are developing a short, intensive course in algebraic model construction. The course combines human teaching with a tutoring system. This paper describes the lessons learned during the iterative development process. Starting from an existing theory of model construction, we gradually acquired a completely different view of the skills required as we modified the tutoring system and the instruction. We close by describing encouraging results from a quasi-experimental study.

Corresponding author: Kurt.vanlehn@asu.edu

Keywords: model construction, intelligent tutoring system, algebra word problems, mathematics learning

Acknowledgements: This research is supported by the US National Science Foundation Grant IIS-1628782.

We gratefully acknowledge the help of Ritesh Samala, Sara Loucks and Swarnalakshmi Lakshmanan.

1 Introduction

Teaching students how to construct and use models is undeniably important. According to the Next Generation Science Standards (NGSS, 2013), "developing and using models" is one of 8 key scientific practices. According to the Common Core State Standards for Mathematics (CCSSM) (NGA & CCSSO, 2011), "modeling with mathematics" is one of its 8 key mathematical practices.

Model construction, and solving algebra word problems in particular, are notoriously difficult for students. A 2007 survey of 743 high school algebra teachers rated word problem solving as the most difficult topic for incoming students (Hoffer, Venkataram, Hedberg, & Shagle, 2007).

According to the CCSSM (NGA & CCSSO, 2011), the process of constructing a model has four sub-processes, shown in Figure 1. When the model is a set of algebra equations, the Formulate process is writing the equations, the Compute process is solving them, and the other two processes check the numerical answer. The Compute process can be performed by a Computer Algebra System (CAS). A CAS will solve equations entered by students. CAS are available on calculators (e.g., Casio 9970Gs or TI-92) as well as computers. Algebra students are often allowed to use a graphing calculator during exams but not a CAS. On the other hand, students in university engineering and science classes are often allowed to use a CAS and sometimes even required to use one. Although a CAS can perform the Compute process of Figure 1, it takes a human to do the other processes. Thus, the non-Compute processes are likely to remain important skills throughout the 21st century. To assess just those skills, our assessments have students use a CAS for the Compute process.

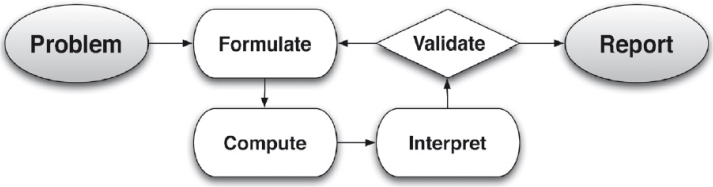

Figure 1: Model construction, from CCSSM, pg. 72

1.1 Prior work on teaching model construction

Many methods for teaching model construction have been investigated (see VanLehn (2013) for a review). Some are general-purpose teaching methods, such as:

- Adaptive task selection (Arroyo, 2000; Beal, Arroyo, Cohen, & Woolf, 2010; Beck, Woolf, & Beal, 2000; Koedinger, Alibali, & Nathan, 2008).
- Self-explanation of worked examples (Cooper & Sweller, 1987; Corbett, Wagner, Lesgold, Ulrich, & Stevens, 2006; Corbett, Wagner, & Raspat, 2003; Heffernan, Koedinger, & Razzaq, 2008; Renkl, Stark, Gruber, & Mandl, 1998).
- Feedback/hints on the solution/model (Biswas, Leelawong, Schwartz, & Vye, 2005; Bravo, van Joolingen, & de Jong, 2009; Zhang et al., 2014).

- Feedback/hints on the student's process (meta-tutoring) (Arnau, Arevalillo-Herraez, Puig, & Gonzalez-Calero, 2013; Chi & VanLehn, 2008, 2010; Leelawong & Biswas, 2008; Zhang et al., 2014)
- Reflective debriefings (Connelly & Katz, 2009; Katz, Allbritton, & Connelly, 2003; Katz, Connelly, & Wilson, 2007).
- Answering student questions (Anthony, Corbett, Wagner, Stevens, & Koedinger, 2004; Beek, Bredeweg, & Lautour, 2011; Corbett et al., 2005; Leelawong & Biswas, 2008; Segedy, Kinnebrew, & Biswas, 2012).
- Students explaining their model (Heffernan et al., 2008; Metcalf, 1999; Metcalf, Krajcik, & Soloway, 2000)
- Gradual increase in model complexity (de Jong et al., 1999; Quinn & Alessi, 1994; Swaak, van Joolingen, & de Jong, 1998; White, 1984, 1993; White & Frederiksen, 1990).
- Gamification (Schwartz et al., 2009).
- Teachable agents and reciprocal teaching (Biswas et al., 2005; Chan & Chou, 1997; Chase, Chin, Oppenzzo, & Schwartz, 2009; Pareto, Arvemo, Dahl, Haake, & Gulz, 2011; Reif & Scott, 1999).

However, model construction seems to be more difficult for students to learn than other STEM skills, so it makes sense to look for teaching methods that are specific to it. Thus, let us briefly review the current accepted theory of how models are constructed.

1.2 The cognitive science of algebra word problems solving

Much fundamental research has been done on the cognitive processes of algebra word problem solving. This section summarizes that work briefly. Reed (1998) provides a summary of the earlier literature. Darocyz, Wolska, Meurers, and Nuerk (2015) summarizes work on *arithmetic* word problems.

Construction of algebraic models has a widely cited theory (Cummins, Kintsch, Reusser, & Weimer, 1988; Fuchs, Fuchs, Seethaler, & Barnes, 2019; Kintsch & Greeno, 1985; Nathan, Kintsch, & Young, 1992). It posits three levels of mental representation:

> In our view, the process of understanding and solving word problems involves three mutually constraining levels of representation that must be constructed by the student: (a) a representation of the textual input itself—the textbase, (b) a model of the situation conveyed by the text in everyday terms—the so-called situation model, and (c) the formalization of that situation—the problem model. (Nathan et al., 1992, pg. 332)

The mapping between the textbase and the situation model is part of the students' general reading ability. However, the mapping between the situation model and the problem model is unique to algebra word problems, and thus will be reviewed here.

The theory posits that people create a problem model by matching *schemas* against the situation model (Fuchs et al., 2019; Fuchs et al., 2010; Jitendra et al., 2015; Marshall, 1995; Nathan et al., 1992; Riley & Greeno, 1988; Xin, Jitendra, & Deatline-Buchman, 2005). Although the term "schema" is used for many kinds of knowledge, the schemas posited here are templates that match against the situation model and insert formal mathematical relationships into the problem model. For the problem of Figure 2, a distance-rate-time (DRT) schema is applied twice: to the plane's trip and to the missile's trip.

Shortly after an F-35 fighter jet passes over some militants, they fire an FIM-92 Stinger missile at the plane. The plane flies at full speed, 537 m/s. The missile flies at its full speed, 750 m/s. How long will it take the missile to catch up with the plane? Assume the missile travels for 6 seconds less than the plane. Ignore the fact that a Stinger missile runs out of fuel after 10 seconds.

Figure 2: A word problem

Nathan et al. (1992) found that students easily identified some relationships but had difficulty identifying others. Nathan et al. (1992) named the easily recognized ones the *governing* relationships of the problem. Figure 2 illustrates all three methods of inserting relationships into the problem model and hence the equations:

```
Dplane = 537 * Tplane          ; a governing relationship – applying DRT to the plane
Dmissile = 750 * Tmissile      ; a governing relationship – applying DRT to the missile
Dplane = Dmissile              ; a supporting relationship – the missile catches up with the plane
Tplane = Tmissile + 6          ; a supporting relationship – a comparison of the flight times
```

These four relationships are *atomic*, that is, they cannot be decomposed into smaller meaningful relationships. A schema that matches an atomic relationship is called an atomic schema. DRT is an atomic schema. The schema for the area of a circle, $A=\pi*r^2$, is also atomic because dividing it requires meaningless "schemas" such as $A=\pi*X$ and $X=r^2$.

However, people who have solved many algebra word problems often have non-atomic schemas that match whole problems. Hinsley, Hayes, and Simon (1977) found that after reading just a fifth of the system description, half the students correctly identified the rest of the problem and its solution. For instance, after hearing just "A river steamer…" one participant said, "It is going to be a linear algebra problem of the current type – it takes four hours to go upstream and two hours to go downstream." (pg. 97).

Although experts know more non-atomic, whole-problem schemas than novices, it appears to be better to teach only atomic schemas. Gerjets, Scheiter and Catrambone (2004, 2006) compared two methods for teaching probability problem-solving. One method matched a schema to the whole problem, classifying it as either a permutation or combination problem. The other method taught students to decompose the system into atomic relationships. Across 7 studies, teaching the decomposition method caused greater learning than teaching students to categorize a whole problem. Blessing and Ross (1996) showed that gifted high school students could be easily fooled into using the wrong whole-problem schema, but would succeed when encouraged to use atomic schemas.

In short, cognitive science indicates that to become skilled in model construction, students must become skilled in decomposing a situation model into atomic relationships such that each matches an atomic schema. However, there is a second skill that must be mastered as well (Nathan et al., 1992). Having written some equations, solvers must recall what the equations, variables and expressions refer to in the situation model. This is sometimes referred to as *grounding* (VanLehn, 2013). For example, in solving the problem of Figure 2, a solver might represent the two governing relationships as X= 537*Y and Z=750*W. It would be impossible to discover that X=Z without recalling that those two variables represent the distances flown by the plane and the missile. In short, when the project began, it appeared that teaching model construction required teaching the four skills shown in Figure 3.

Decomposing a situation model into relationships that match atomic schemas

1. Recognizing governing relationships.
2. Recognizing supporting relationships.

Grounding, that is, maintaining awareness of…

3. what quantities the model's variables and expressions refer to.
4. what relationships the model's equations refer to.

Figure 3: The four critical skills of model construction

1.3 Theory-based teaching methods

Several methods of teaching model construction are based on the theory sketched above. Perhaps the most well-developed method teaches arithmetic (not algebraic) model construction via direct instruction (Fuchs, Fuchs, Finelli, Courey, & Hamlett, 2004; Fuchs et al., 2003; Fuchs, Fuchs, Prentice, et al., 2004; Fuchs et al., 2019; Fuchs et al., 2009; Fuchs et al., 2010; Hutchinson, 1993; Jitendra et al., 2007; Jitendra et al., 2015; Jitendra, Star, Dupuis, & Rodiguez, 2013; Jitendra, Star, Rodriguez, Lindell, & Someki, 2011; Jitendra et al., 2009; Xin et al., 2005; Xin et al., 2001). The instructional method is called Schema-Broadening Instruction because students are explicitly taught atomic schemas, and the problem sequences gradually vary or remove surface features in order to promote generalization of the schemas. *Grounding* of variables is enhanced both by using descriptive variable names instead of the traditional single letter and by highlighting words in the problem statements. *Grounding* of mathematical relationships is enhanced by using node-link diagrams in addition to equations. *Decomposition* is not usually addressed because most investigations targeted young and special-education populations using simple problems that could all be solved by applying a single atomic schema.

Another method with ample research helps students ground algebraic equations by having them first solve several numerical versions of the problem (Heffernan, 2003; Heffernan & Croteau, 2004; Heffernan & Koedinger, 1997; Heffernan et al., 2008; Koedinger & Anderson, 1998; McArthur et al., 1989). For example, to solve this problem,

Drane & Route Plumbing Co. charges $42 per hour plus $35 for the service call. Find the number of

hours worked when you know the bill came out to $140.

the method first asks the student to calculate the bill when the plumbers worked 3 hours, and again when they worked 4.5 hours. Finally, it asks them to write an equation using "x" for the number of hours worked. This method fared well in several evaluations (op. cit.) and is used in commercial algebra tutoring systems (Carnegie Learning, 2020). However, it does not support decomposition into atomic schemas.

Grounding was also the focus of several other teaching methods. Some methods augmented variables with icons or thumbnail images (Avouris, Margaritis, Komis, Saez, & Melendez, 2003; Forbus, Carney, Sherin, & Ureel II, 2005; Metcalf et al., 2000). Others replaced equations with bar graphs (Looi & Tan, 1996, 1998; Munez, Orrantia, & Rosales, 2013) or node-link diagrams (Bridewell, Sanchez, Langley, & Billman, 2006; Chang, Sung, & Lin, 2006; Derry & Hawkes, 1993; Löhner, Van Joolingen, & Savelsbergh, 2003; Löhner, Van Joolingen, Savelsbergh, & Van Hout-Wolters, 2005; Marshall, 1995; McArthur et al., 1989; Metcalf et al., 2000; Pauli & Reusser, 1997; Reusser, 1993; van Joolingen, De Jong, Lazonder, Savelsbergh, & Manlove, 2005; Willis & Fuson, 1988). Animations have been used to help students ground the whole model (Gould & Finzer, 1982; Nathan et al., 1992).

Only a few methods have focused on decomposition (Heffernan & Koedinger, 1997; Heffernan et al., 2008; Ramachandran, 2003). In particular, one study suggested that teaching decomposition would be more effective than methods focused on aiding grounding (Heffernan & Koedinger, 1997).

Two methodologies were prominent in prior research. One was the traditional vary-one-thing-at-a-time method, which compared two methods of instruction that differ only in a single aspect (e.g., with vs. without animation). The other methodology is often called design-based research (Reimann, 2011) or iterative development. The researchers start with instruction that is based on several ideas, then change the instruction based on feedback from teachers and students as the method is used in classrooms. Because it sometimes takes time to process the classroom data and make changes in the method, the work is often structured as a sequence of formative evaluations alternating with analysis of the data and modification of the instruction. The basic idea is hill-climbing: making small changes in order to evolve the instruction from an initial version to a potentially optimal version. Evidence from a formative evaluation is usually too weak to support publication. It may consist of statistically insignificant trends in the data, conversations with teachers, and/or informal observation and interaction with students. Some of the changes to the instruction are so small and specific that they would not interest most readers. Nonetheless, this methodology has created some highly effective instruction (Reimann, 2011).

In our earlier research on model construction in physics (VanLehn et al., 2007; VanLehn et al., 2005) and system dynamics (VanLehn, Chung, Grover, Madni, & Wetzel, 2016; VanLehn, Wetzel, Grover, & van de Sande, 2017; Wetzel et al., 2016), we noticed that although the overall impact of our tutoring systems was positive, some students learned little. Inspired by the success of Schema-Broadening Instruction for similar under-performing populations, we set about using design-based research to develop a tutoring system for algebraic

model construction. Moreover, because physics and dynamic systems models often require many atomic schemas, we decided to focus on multi-schema problems and the decomposition skill.

This publication reports our progress. It tells our story chronologically, from the initial formative evaluations, the initial version of the tutoring system, changes made during subsequent formative evaluations and our first summative evaluation. This story is aimed at two audiences. For those who are also working on teaching model construction, the insights we gained from the formative evaluations may be valuable. For those who are iteratively developing their own instructional systems, seeing how our iterative development went may help them hone their own methodology. The summative evaluation uses a regression discontinuity design, which is unusual in the AI and Education literature; readers may it find interesting.

2 Developing the initial system

Our formative evaluations all consisted of a multi-hour "boot camp" as a supplementary instruction, given outside a regular college algebra class. Although the instruction could be eventually incorporated into a course, running it as a supplement allowed us to observe and interact more closely with the students.

The boot camp was developed iteratively, alternating between formative (i.e., diagnostic) evaluations and instruction redesign. The first formative evaluation (Spring 2017) used paper instead of software. Four students completed our 20-hour boot camp. They met as a small class with an instructor (Banerjee) and the other authors of this paper as instructional assistants. We tried three types of notation for models: traditional algebraic equations, tables and node-link networks. We tried several types of strategic scaffolding, discussed in the next section. Because there were 4 students and 4 experimenters in the room, much was learned during conversations and close observation. Over the next year, we modified a tutoring system developed for an earlier project (VanLehn et al., 2016; VanLehn et al., 2017; Wetzel et al., 2016). The new system was named TopoMath.

3 Initial version of Topomath

This section describes the major features of the initial version of TopoMath.

3.1 TopoMath's notation facilitates grounding

Ordinary algebra notation is optimized for handwriting, because expressions are often copied many times during the process of solving them. For instance, variable names are often a single letter, and multiple relationships are expressed in a single equation. This brevity impedes grounding. Students have difficulty remembering what short variable names stands for. They have even more difficulty identifying what algebraic expressions denote (Corbett et al., 2006). For example, suppose students use DPlane = 537 * (TMissile + 6) in their model of Figure 2. They may have trouble remembering that (TMissile + 6) represents the duration of the plane's flight. When a CAS does the solving, such brevity is no longer necessary.

Thus, we developed a notation that facilitates grounding. It is based on notations used by MathCAL (Chang et al., 2006), Heron (Pauli & Reusser, 1997; Reusser, 1993, 1996) and Schema-Broadening Instruction (Fuchs et al., 2019; Fuchs et al., 2009; Fuchs et al., 2010; Jitendra et al., 2015; Jitendra et al., 2013; Jitendra et al., 2011;

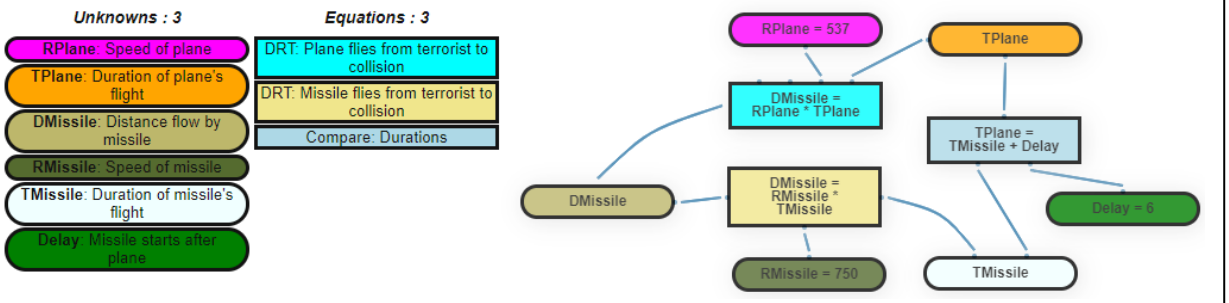

Figure 4: TopoMath model for problem of Figure 2. Descriptions on left; network on right.

Jitendra et al., 2009). The basic idea is to use a network of nodes that represent variables and equations. The variable nodes have long names inside them, which facilitates remembering what the variable denotes. The equation nodes have only equations from atomic schema applications. This means there are no algebraic expressions that denote quantities, avoiding the grounding difficulties studied by Corbett et al. (2006)

Figure 4 presents a model for the problem discussed earlier. The rectangular nodes (called equation nodes) contain an equation representing an atomic relationship. The ovals (called variable nodes) represent a quantity. When a quantity's value is given in the problem statement (i.e., it is a parameter of the model), then the variable node displays that value.

TopoMath also facilitates grounding by listing longer descriptions of both variable nodes and equation nodes in columns on the left. The colors match those in the network.

3.2 Schema menu

As Mayer (1981) pointed out, there are many atomic schemas such as DRT, the area of a circle, the ideal gas law P*V=n*R*T and Newton's 2nd law F=m*a. Since we are not interested in teaching any particular atomic schema, TopoMath provides students with a menu of atomic schemas. When they select one and fill its slots with quantities, TopoMath creates an equation node and writes the equation inside it. This allows TopoMath and the instructors to focus on the key skills of decomposition and grounding.

3.3 Other features of TopoMath

TopoMath is a step-based intelligent tutoring systems that uses example-tracing to determine if a step is correct or incorrect (VanLehn, 2006, 2011). A step consists of a menu selection, typing into a box, or other simple user interface acts. TopoMath gives immediate feedback on steps but does not give hints. A step turns green if it is correct or red if it is incorrect. On the third incorrect attempt, TopoMath does the step correctly and colors it yellow. Instructors encourage students to self-explain yellow steps (Shih, Koedinger, & Scheines, 2008).

Defining an equation node or a variable node takes several steps executed in a pop-up editor window. When the window is closed, if any step is yellow, then the node has a yellow icon so that anyone looking at the model can tell that the student may have been abusing the feedback (Baker, Corbett, Roll, & Koedinger, 2008).

On the other hand, if all the node's steps were completed correctly on the first attempt, the node gets a gold star icon.

The overall TopoMath system includes an authoring system, a real-time dashboard and an LCMS. Instructors have several options for controlling e.g., when TopoMath gives feedback and what nodes are given to students initially.

4 Formative evaluations of TopoMath

This section briefly describes the remaining formative evaluations. The next section describes what we learned from them.

The first formative evaluation of TopoMath (Spring 2018) had 10 students working as a small class for 20 hours with the authors present. Again, we learned much via conversations and close observation.

In fall 2018, a revised version of TopoMath was used in a 50-student ASU class. Although we did not attempt detailed interpersonal interactions, we uncovered many software bugs and received several suggestions for improvements.

In all formative evaluations, students varied dramatically in how fast they learned. For the class-paced boot camps of Spring 2017 and 2018, this meant that fast learners were bored and slow learners were confused because the whole class instruction was paced for average-speed learners. For the boot camp of Spring 2019, we used self-paced learning. Instead of having students meet as a class, we had them sign up for time slots and work in our lab so that they could move at their own pace. A human tutor was always present. Despite financial incentives, only 3 of the 8 students completed the 20-hour intervention. Students who dropped out often complained about the "hassle" of coming to the lab.

The boot camp was revised one last time. In addition to several modifications to accommodate lessons learned (described in the next section), we prepared for a summative evaluation. In order to reduce the time required from 20 hours to roughly 6 hours, we eliminated coverage of several advanced topics. We reconfigured the intervention to be a short online course on Canvas, the learning management system used at our school, so that students could work at home, in the lab or in a classroom. This is the version of the TopoMath boot camp whose summative evaluation is presented later.

5 Lessons learned from formative evaluations of TopoMath

This section lists revisions to our initial view of the skill of algebraic model construction. These insights were gleaned mostly by observing and talking with students during the formative evaluations of TopoMath.

5.1 Grounding governing schema applications

Recall that governing relationships are the ones that students easily recognize. Their recognition appears to use surface features, which is consistent with the wider literature on schema recognition (Reed, 1998). For example, in Figure 2, they can easily recognize that DRT is relevant because the problem mentions speed, seconds, and other surface features.

When problems are simple, surface features often suffice. But when a problem has two or more occurrences of the same schema, then students get confused. TopoMath (and all other schema editors) requires that students enter something that will distinguish one schema application from the others. The indication does not have to be long, but it does have to be unambiguous. In Figure 3, one schema application is referred to as "DRT applied to the missile" and the other is referred to as "DRT applied to the plane."

We first considered having students type in such grounding descriptions. However, in talking to students, we found that they had difficulty describing and distinguishing such concrete, grounding relationships. In general, they had difficulty speaking precisely.

However, it seemed to us that they had the appropriate concepts but could not easily put them into words. When we listed possible relationships, they rapidly recognized the appropriate one. Thus, TopoMath now has students pick a relationship from a menu provided by the author of the problem.

5.2 Recognizing supporting relationships

Supporting relationships are the ones that are difficult for students to recognize. We tried several instructional methods before finding one that worked.

We tried teaching explicit schemas and rules for when to apply them. For example, one rule was, "If two moving objects, A and B, cover the same distance and end at the same time, but object B starts later then object A, then the travel time of A equals the travel time of B plus the delay." This rule would be used to construct the "compare durations" schema application of Figure 2. We considered each rule to be a knowledge component (Koedinger, Corbett, & Perfetti, 2012), and envisioned teaching dozens of them. However, students struggled to understand the language of the rules. When they finally understood, they said they knew the rule already, so they did not know why we were bothering to teach it. Thus, we gave up teaching such rules. The students' problem was not lack of knowledge, it was activating the knowledge appropriately. They needed recognition cues.

The Target Variable Strategy (Chi & VanLehn, 2010; Vanlehn & Chi, 2012) is goal recursion applied to algebraic model construction. After writing an equation, one identifies unknowns in it and then sets the goal of finding a new equation that contains one of the unknowns. This method worked well in earlier studies (op. cit.), but failed in our context. In particular, it forced students to try to find a supporting relationship when they had not yet represented all the governing relationships. Students complained that they wanted to write a governing relationship but the tutoring system would not let them. Thus, we decided to let students represent relationships in any order they wanted.

Finally, we discovered an easily taught heuristic for cueing recognition of supporting relationships. We made the nodes movable and taught students that *whenever they get stuck, they should first locate all unknown variable nodes that have just one line connected to them, and then drag such nodes into clusters of similar nodes.* Using Figure 2 as an example, suppose students have noticed only the two governing relationships and created the two equation nodes Dmissile = Rmissile*Tmissile and Dplane=Rplane*Tplane. Suppose they are unable to identify the two supporting relationships. Thus, the heuristic has them drag the two distance nodes (Dmissile and Dplane) close together and the two duration nodes (Tmissile and Tplane) close together (see Figure 5). This visually invites connecting the nodes somehow, which leads to the discovery of the missing relationships, Dmissile=Dplane and Tplane = Tmissile + Tdelay.

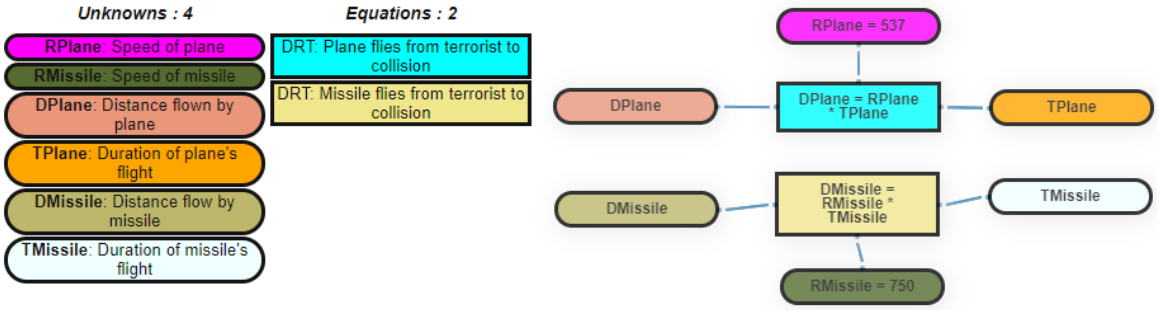

Figure 5: Grouping similar unknowns

5.3 Grounding variables and filling slots

Grounding a variable means maintaining awareness of what quantity in the situation model the variable represents. Our first efforts focused on making variable names unambiguous to students.

One approach we tried orally was to ask students to give their variables unambiguous names. We gave them feedback if the name was not precise enough. Students found this frustrating. For a problem involving original price, price after a discount, and final price after both tax and discount, they might try first X, then P, then Price, then DiscountPrice and finally get really irritated at the human tutor who was giving them feedback. They told us it was like trying to read the mind of the tutor.

We tried using short variable names and displaying a definition of each so that students would not need to tax their memory. This failed, because students focused only on the equations where the short names appeared and rarely seemed to consult the list of definitions. When we asked them what such a variable, e.g., D1, referred to, they could not immediately say.

We also tried the opposite extreme, using long descriptive variable names. When constructing a model, students could choose from a menu of such names. Unfortunately, students typically would read only a few words in the menu item, so they overlooked details. For example, they might choose “the amount of tax in dollars” because it was the first item in the menu with the word “tax” in it even though “tax rate as a proportion” appeared later and was the appropriate choice. So, this approach failed as well.

We considered highlighting the text in the system description that referred to quantities, then having students select a highlighted text as part of the process of defining a variable. However, we discovered that some quantities are not mentioned anywhere in the text. For example, Figure 2 does not mention distances.

Sometimes students do understand exactly what all the variables refer to, but they still have trouble constructing a model. For example, when they had constructed the model of Figure 5, they might still have trouble recognizing that DMissile and DPlane were same. Indeed, it could be argued that "the distance traveled by the missile" is different from "the distance traveled by the plane," but the *values* of the two quantities were equal. Deciding whether two quantities are "the same" can be quite confusing. This may be related to the broader problem of students' weak and varying conceptions of the equals sign (Simsek, Xenidou-Deryou, Karadeniz, & Jones, 2019), and to the old philosophy puzzle: do "the evening star," "the morning star" and "Venus" all refer to same thing?

Finally, we realized that it was not necessary to view variables as referring to quantities. Learners could reason with the slots of schemas instead. For example, instead of first determining what "DMissile" refers to in the situation model, then determining what "DPlane" refers to, and then deciding that their referents are actually the same thing, we just ask students to decide whether the distance slot of the schema DRT:Missile should be filled with the same variable as the distance slot of the schema DRT:Plane. On this view, a variable is either parameter (i.e., its value is given in the problem) or an indicator that two slots refer to the same quantity. This ducks the whole issue of quantities, grounding variables and determining whether two quantities were "the same." This was a major change in our thinking about model construction problem solving. A philosopher would say that co-reference has replaced nominal reference in our epistemology. Of the four skills that need to be taught (Figure 3), the third skill becomes, "Determine which slots co-refer." This change in our thinking precipitated a major change in the TopoMath user experience.

The initial version of TopoMath had a "Create Variable" button and a "Create Equation" button. Both created nodes. We removed the "Create Variable" button and modified the equation node editor as follows. After the student has chosen a description for the relationship (e.g., either "DRT applied to the plane" or "DRT applied to the missile"), TopoMath fills the slots of schema with default variable names (see Figure 6). A default variable name is a concatenation of slot-identifying letters (D, R or T in the case of the DRT schema) and a short form of the relationship (e.g., "Plane" or "Missile"). Thus, the default variable name labels a slot unambiguously. When the student closes the equation node, new variable nodes are created as needed. These are as shown in gray (see Figure 7, upper part), indicating that they are not yet complete.

The student must click on a gray variable node in order to finish its definition. As shown in Figure 7, lower half, students must decide whether the variable is a parameter or not, and if it is a parameter, they must type in its value, which is specified in the system description.

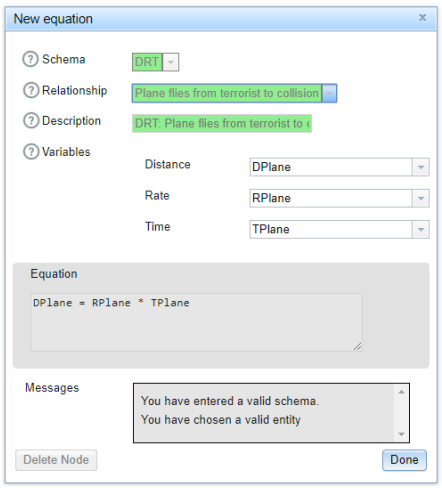

Figure 6: Default names fill the slots

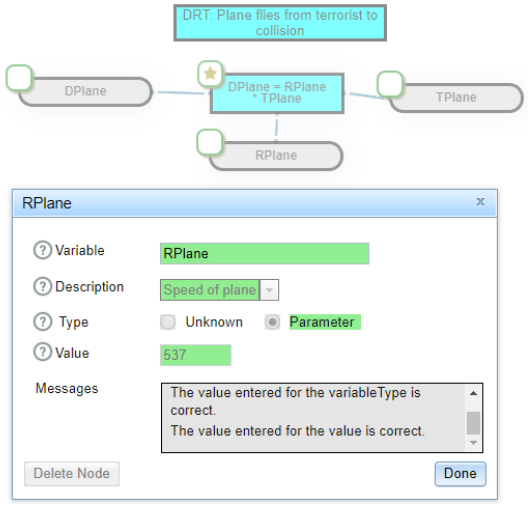

Figure 7: Finishing up a variable node

If students decide later that two slots should be filled with the same variable, then they can edit one of the equation's nodes slots so that it is filled with the variable denoting the other slot. When they close the node editor, TopoMath eliminates the now-superfluous variable node. For example, suppose the model is in the state shown in Figure 5 and students decide that the two distances are equal. They can open DRT:Missile and use the drop down menu to fill its distance slot with DPlane. When they click the Done button, the node for DMissile is deleted and the left part of the model looks like Figure 4. Alternatively, then can edit DRT:Plane to fill its distance slot with DMissile.

TopoMath also allows students to skip some of these steps if they realize early that two schemas' slots co-refer. For example, suppose they have already created DRT:Missile, they are now creating DRT:Plane, and they realize that the two schemas refer to the same distance. To fill the distance slot of DRT:Plane, they replace its default filler, Dplane, by selecting DMissile from a menu of previously defined variables.

To summarize: If students immediately notice that two slots co-refer, then they can get the same variable to fill both slots by using the menu of existing variables. If they do not immediately notice the co-reference, then they can retain the default variable names, close the equation node, use the node-clustering heuristic to discover the co-reference, and then edit the model to make the slots share a variable.

6 First summative evaluation

This section presents a summative evaluation of the TopoMath boot camp. Whereas a formative evaluation is intended to maximize the insights the experimenters can get from close interaction with the students, a summative evaluation is intended to determine whether an intervention "works well." Because the boot camp is intended to supplement regular instruction, this means showing that students who take the boot camp learn more than students who do not. Unlike a formative evaluation, experimenters should not change the instruction in the middle of a summative evaluation.

6.1 Subjects and setting

The study took place in the first few weeks of a university class that taught students how to construct computational models in a variety of formalisms, including mathematical models, probabilistic models, sequential event models and agent-based models. Most students were in their junior or senior year of an engineering major. Nonetheless, prior experience with this class suggested that students varied widely in their mathematical competence. Some could not solve even the simplest algebra word problems, while others were comfortable with calculus and differential equations. Of the 75 students enrolled in the course, 62 consented to have their data used in this study.

6.2 Study design

Because the competence of students varied widely, the study used a regression discontinuity design. On the second day of class, students took a placement test. Students who scored below a cutoff were required to take our algebra word problem boot camp. Those who scored above the cutoff were not required to take the boot camp, but were allowed to do so. When the boot camp was finished, all students took a midterm exam that assessed their competence at constructing algebraic models. When a regression discontinuity design works, the regression of post-test score against pre-test score has a discontinuity at the cutoff. This discontinuity will be illustrated in our results section.

6.3 Measures

The 12-problem placement test and the midterm were isomorphic. That is, they had similar problems in the same order. Only the cover stories and numbers changed. After students finished the placement test, which served as the pre-test, they could see neither the test nor their marks.

The tests were taken on a conventional CAS, where students enter equations in the standard notation. To prevent students from solving the problems in ad hoc ways on paper and just entering the answer, the CAS allowed them to enter only parameter values; all other numbers were banned from equations. Before taking the pre-test, students spent 10 minutes learning how to use this CAS. Students were told the scoring rubric, which as simple: If one of the variables in their model was assigned the correct answer's value, then that problem was marked 100% correct; otherwise, it was marked 0% correct. Partial credit was not given.

The problems ranged from simple, single-relationship problems up to problems with 5 atomic relationships. Problems were adapted from a popular college algebra textbook. The reliability, validity and equality of the tests have not yet been measured.

6.4 Procedure

On the second day of class, all students took the pre-test (= placement test). The test was limited to one hour. The placement score cutoff was set so that the top third of the students were above cutoff. Student who scored below the cutoff (which was 61%, by the way) were required to complete the TopoMath boot camp, which was a series of lessons on Canvas. They were required to complete it before the post-test (= midterm exam) working at their own speed. They were also required to attend 3 classes, which were run as

study halls. Although they were encouraged to collaborate with each other during class and to ask questions of the instructor, the study halls were rather quiet. Students were also encouraged to come to office hours, and one student did. The midterm exam occurred 2½ weeks after the placement exam.

6.5 Results

Although a third (22) of the consenting students scored above the cutoff and were thus allowed to skip the boot camp, only 5 students skipped it entirely. The other 17 took some or all of TopoMath boot camp. This creates a potential self-selection bias. To test for such a bias, we compared the pre-test scores of both groups. Those who took TopoMath scored 17.3 whereas those who skipped TopoMath scored 16.6, an unreliable difference ($p=0.38$). Their gains from pre-test to post-test were also not reliably different ($p=0.90$). Thus, despite the potential self-selection bias, we treated the 5 students who skipped TopoMath as the no-treatment group.

Among the 40 students who were below cutoff, all completed almost all the boot camp, so we included them all in the treatment group. Thus, there were 40 students in the treatment group and 5 in the no-treatment group.

Figure 8 scatterplots the results, with some points representing two or more people. The no-treatment group showed a clear linear relationship between their test scores, which is to be expected given that the tests were isomorphic and the students received no treatment between the tests. The effect of the treatment was to raise the regression line of the treatment group so that their post-test score was about 10% higher than would be predicted by their pre-test score if they had not participated in the algebra boot camp.

However, the visually apparent regression discontinuity may not be statistically reliable given the small number of students in the no-treatment condition. To test the hypothesis that a two-line, discontinuous model fits better than a model with just one regression line, we defined a variable, PreTest, by subtracting the cutoff value, 0.61, from the placement test scores of all the students. We also included a Condition variable that was 0 for students in the treatment group and 1 for students in the no-treatment group. We then fit this linear model:

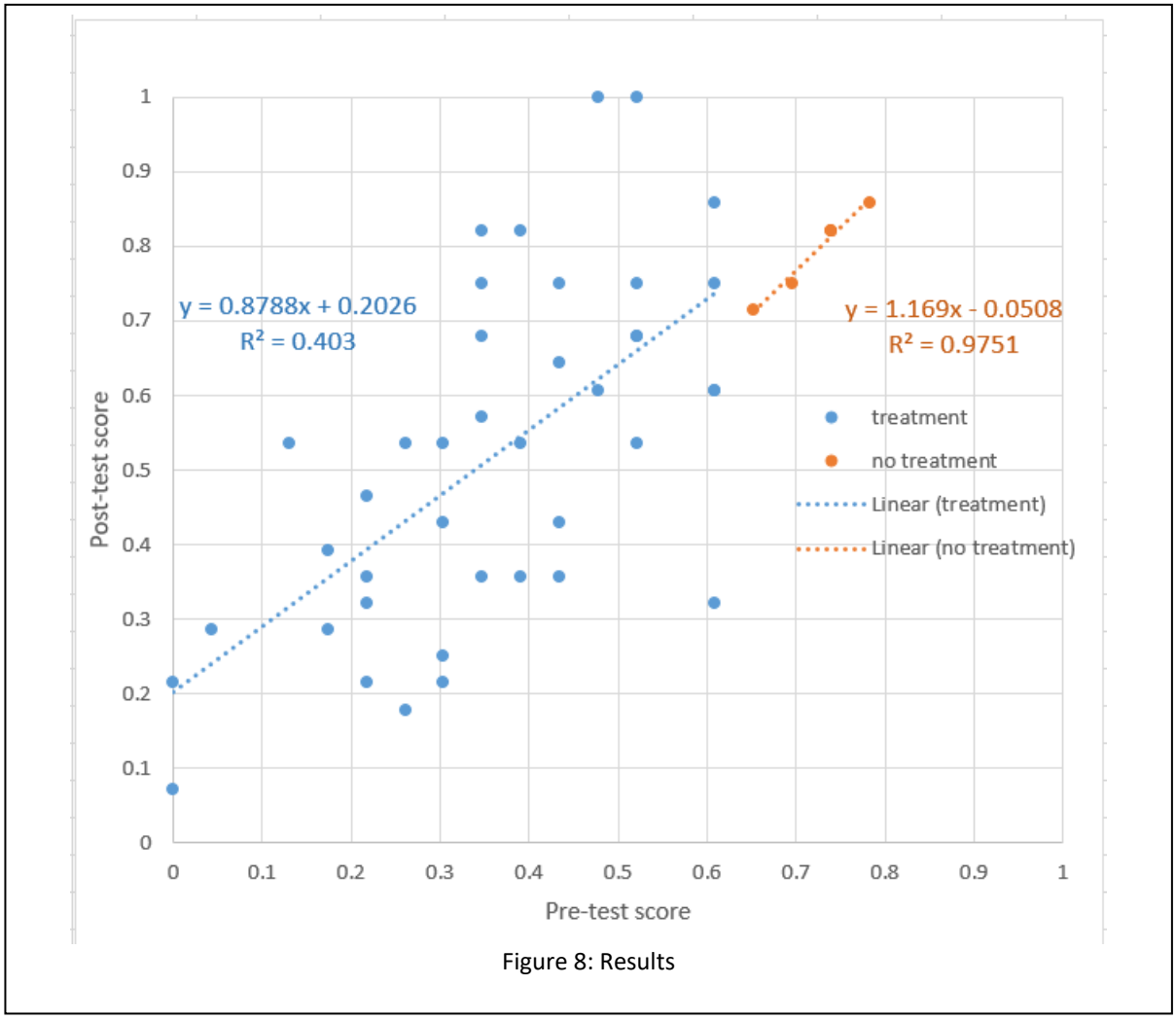

Figure 8: Results

PostTest = $\beta_0 + \beta_1$*PreTest + β_2*Condition + β_3*Condition*PreTest

If the condition made a significant difference, then the last two terms should contribute reliably to explaining the variance in the PostTest score, because they essentially establish the intercept and slope of a second regression line. However, neither of those terms contributed reliably to the fit ($p=0.74$ for β_2 and $p=0.87$ for β_3). The two-line model remained unreliable when we augmented the no-treatment group with the 17 students who used TopoMath voluntarily. In other words, a model with two regression lines (as shown in Figure 7) did *not* fit better than a model with just one regression line.

7 Discussion

Although Schema Broadening Instruction has a large effect for *arithmetic* story problems (Fuchs et al., 2009; Fuchs et al., 2010; Jitendra et al., 2013; Jitendra et al., 2011; Jitendra et al., 2009), no similar large effect has been demonstrated for *algebra* story problems. In order to develop effective algebraic modeling instruction, we employed an iterative development or design-based research methodology. We hope the story of our development will encourage others to share their stories as well.

During our formative evaluations, we manipulated many features and collected qualitative impressions, mostly from talking with students. The main insights were: (1) When students solve problems with more than one application of the same schema, they have difficulty describing the differences between the schema applications but are able to easily recognize the appropriate author-written descriptions. (2) Instead of trying to get students to ground variables (i.e., understand unambiguously what quantity the variable denotes), it is better to help them discover slots that share a variable. (3) When students have trouble recognizing supporting relationships or variable sharing, grouping similar variables helps them. These insights are typical of those from iterative development in that they are specific to the teaching task and mostly of interest to those addressing the same teaching task.

Our summative evaluation suggests that we might be starting to move the needle, but we still have a long way to go. Although the regression line of the treated students was higher than the regression line of the no-treatment students, the discontinuity was too small to be reliable in the context of the large unexplained variance that is typical of classroom studies.

Because the summative evaluation provided little information on how to change TopoMath, we have gone back to formative evaluations. We are in the midst of preparing a new version of TopoMath, and planning on another summative evaluation in the fall.

For summative evaluations, there is probably no experimental design that compares a treatment group to a no-treatment group, and yet allows all students equal access to the treatment as required of equitable classroom instruction. Our regression discontinuity design was flawed because most of the above-cutoff group students used TopoMath even though they were not required to do so. This reduced the size of the no-treatment group to 5 and created a selection confound: the students themselves deciding whether to be in the no-treatment group. To avoid this during the next summative evaluation, we will prevent the above-cutoff students from using TopoMath between the pre-test and post-test, thus all the above-cutoff students will be in the no-treatment group. However, it would be unethical to deny the no-treatment students instruction if they want it, so if they do not like their score on the post-test, we will give the them access to TopoMath and a second exam. Their score on the second exam will be used for grading purposes but will not be used in the study. This design has flaws too, but nonetheless seems an improvement over the standard regression discontinuity design.

References

Anthony, L., Corbett, A. T., Wagner, A. Z., Stevens, S. M., & Koedinger, K. R. (2004). Student question-asking patterns in an intelligent algebra tutor. In J. C. Lester, R. M. Vicari, & F. Praguacu (Eds.), *Intelligent Tutoring Systems: 7th International Conference, ITS 2004* (pp. 455-467). Berlin: Springer-Verlag.

Arnau, D., Arevalillo-Herraez, M., Puig, L., & Gonzalez-Calero, J. A. (2013). Fundamentals fo the design and the operation of an intelligent tutoring system for the learning of the arithmetical and algebraic way of solving word problems. *Computer and Education, 63*, 119-130.

Arroyo, I. (2000). *AnimalWatch: An arithmetic ITS for elementary and middle school students*. Paper presented at the Workshop at ITS 2000.

Avouris, N., Margaritis, M., Komis, V., Saez, A., & Melendez, R. (2003). *ModellingSpace: Interaction design and architecture of a collaborative modelling environment*. Paper presented at the Sixth International Conference on Computer Based Learning in Sciences (CBLIS), Nicosia, Cyprus.

Baker, R. S. J. d., Corbett, A., 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.

Beal, C., Arroyo, I., Cohen, P. R., & Woolf, B. P. (2010). Evaluation of AnimalWatch: An intelligent tutoring system for arithmetic and fractions. *Journal of Interactive Online Learning, 9*(1), 64-77.

Beck, J., Woolf, B. P., & Beal, C. (2000). ADVISOR: A machine learning architecture for intelligent tutor construction. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence* (pp. 552-557). Menlo Park, CA: AAAI Press.

Beek, W., Bredeweg, B., & Lautour, S. (2011). Context-dependent help for the DynaLearn modelling and simulation workbench In G. Biswas (Ed.), *Artificial Intelligence in Education* (pp. 4200-4422). Berlin: Springer-Verlag.

Biswas, G., Leelawong, K., Schwartz, D. L., & Vye, N. J. (2005). Learning by teaching: A new agent paradigm for educational software. *Applied Artificial Intelligence, 19*, 263-392.

Blessing, S. B., & Ross, B. H. (1996). Content effects in problem categorization and problem solving. *Journal of Experimental Psychology: Learning, Memory and Cognition, 22*(3), 792-810.

Bravo, C., van Joolingen, W. R., & de Jong, T. (2009). Using Co-Lab to build system dynamics models: Students' actions and on-line tutorial advice. *Computer and Education, 53*, 243-251.

Bridewell, W., Sanchez, J. N., Langley, P., & Billman, D. (2006). An interactive environment for the modeling and discovery of scientific knowledge. *international Journal of Human-Computer Studies, 64*, 1099-1114.

Carnegie Learning. (2020). Cognitive Tutors. Retrieved from http://www.carnegielearning.com/

Chan, T.-W., & Chou, C.-Y. (1997). Exploring the design of computer supports for reciprocal tutoring. *International Journal of Artificial Intelligence and Education, 8*, 1-29.

Chang, K.-E., Sung, Y.-T., & Lin, S.-F. (2006). Computer-assisted learning for mathematical problem solving. *Computers & Education, 46*, 140-151.

Chase, C. C., Chin, D. B., Oppenzzo, M., & Schwartz, D. L. (2009). Teachable agents and the Protégé effect: Increasing the effort towards learning. *Journal of Science Education and Technology, 18*(4), 334-352.

Chi, M., & VanLehn, K. (2008). Eliminating the gap between the high and low students through meta-cognitive strategy instruction. In B. P. Woolf, E. Aimeur, R. Nkambou, & S. P. Lajoie (Eds.), *Intelligent Tutoring Systems: 9th International Conference: ITS2008* (pp. 603-613). Berlin: Springer.

Chi, M., & VanLehn, K. (2010). Meta-cognitive strategy instruction in intelligent tutoring systems: How, when and why. *Journal of Educational Technology and Society, 13*(1), 25-39.

Connelly, J., & Katz, S. (2009). Toward more robust learning of physics via reflective dialogue extensions. In G. Siemens & C. Fulford (Eds.), *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications 2009* (pp. 1946-1951). Chesapeake, VA: AACE.

Cooper, G., & Sweller, J. (1987). Effects of schema acquisition and rule automation on mathematical problem-solving transfer. *Journal of Educational Psychology, 79*(4), 347-362.

Corbett, A., Wagner, A. Z., Chao, C.-y., Lesgold, S., Stevens, S. M., & Ulrich, H. (2005). Student questions in a classroom evaluation of the ALPS learning environment. In C.-K. Looi & G. McCalla (Eds.), *Artificial Intelligence in Education* (pp. 780-782). Amsterdam: IOS Press.

Corbett, A., Wagner, A. Z., Lesgold, S., Ulrich, H., & Stevens, S. M. (2006). The impact of learning of generating vs. selecting descriptions in analyzing algebra example solutions. In S. A. Barab, K. E. Hay, & D. T. Hickey (Eds.), *The 7th International Conference of the Learning Sciences* (pp. 99-105). Mahwah, NJ: Erlbaum.

Corbett, A., Wagner, A. Z., & Raspat, J. (2003). The impact of analysing example solutions on problem solving in a pre-algebra tutor. In U. Hoppe, F. Verdejo, & H. Kay (Eds.), *Artificial Intelligence in Education: Proceedings of AIED 2003: The 11th International conference on AI in Education* (pp. 133-140). Washington DC: IOS Press.

Cummins, D. D., Kintsch, W., Reusser, K., & Weimer, R. (1988). The role of understanding in solving word problems. *Cognitive Psychology, 20*, 405-438.

Darocyz, G., Wolska, M., Meurers, W. D., & Nuerk, H.-C. (2015). Word problems: A review of linguistics and numerical factors contributing to their difficulty. *Frontiers in Psychology, 6*, 348-362.

de Jong, T., Martin, E., Zamarro, J.-M., Esquembre, F., Swaak, J., & van Joolingen, W. R. (1999). The integration of computer simulation and learning support: An example from the physics domain of collisions. *Journal of Research in Science Teaching, 36*(5), 597-615.

Derry, S. J., & Hawkes, L. W. (1993). Local cognitive modeling of problem-solving behavior: An application of Fuzzy Theory. In S. P. Lajoie & S. J. Derry (Eds.), *Computers as Cognitive Tools* (pp. 107-140). Hillsdale, NJ: Lawrence Erlbaum Associates.

Forbus, K. D., Carney, K., Sherin, B. L., & Ureel Il, L. C. (2005). VModel: A visual qualitative modeling environment for middle-school students. *AI Magazine, 26*(3), 63-72.

Fuchs, L. S., Fuchs, D., Finelli, R., Courey, S. J., & Hamlett, C. L. (2004). Expanding schema-based transfer instruction to help third graderes solve real-life mathematical problems. *American Education Research Journal, 41*(2), 419-445.

Fuchs, L. S., Fuchs, D., Prentice, K., Burch, M., Hamlett, C. L., Owen, R., . . . Jancek, D. (2003). Explicitly teaching for transfer: Effects on third-grade students' mathematical problem solving. *Journal of Educational Psychology, 95*(2), 293-305.

Fuchs, L. S., Fuchs, D., Prentice, K., Hamlett, C. L., Finelli, R., & Courey, S. J. (2004). Enhancing mathematical problem solving among third-grade students with schema-based instruction. *Journal of Educational Psychology, 96*(4), 635-647.

Fuchs, L. S., Fuchs, D., Seethaler, P. M., & Barnes, M. A. (2019). Addressing the role of working memory in mathematical word-problem solving when designing intervention for struggling learners. *ZDM Mathematics Education*.

Fuchs, L. S., Powell, S. R., Seethaler, P. M., Cirino, P. t., Fletcher, J. M., Fuchs, D., . . . Zumeta, R. O. (2009). Remediating number combinations and word problem deficits among students with mathematics difficulties: A randomized control trial. *Journal of Educational Psychology, 101*(3), 561-576.

Fuchs, L. S., Zumeta, R. O., Schumacher, R. F., Powell, S. R., Seethaler, P. M., Hamlett, C. L., & Fuchs, D. (2010). The effects of schema-broadening instruction on second grader's word-problem performance and their ability to represent word problems with algebric equations: A randomized control study. *The Elementary School Journal, 110*(4), 440-463.

Gerjets, P., Scheiter, K., & Catrambone, R. (2004). Desiging instructional examples to reduce intrinsic cognitive load: Molar versus modular presentation of solution procedures. *Instructional Science, 32*, 33-58.

Gerjets, P., Scheiter, K., & Catrambone, R. (2006). Can learning from molar and modular worked examples be enhanced by providing instructional explanations and prompting self-explanations? *Learning and Instruction, 16*, 104-121.

Gould, L., & Finzer, W. (1982). A study of TRIP: A computer system for animating time-rate-distance problems. *International Journal of Man-Machine Studies, 17*, 109-126.

Heffernan, N. T. (2003). Web-based evaluations showing both cognitive and moitivational benefits of the Ms. Lindquist tutor. In *Proceedings of the 11th International Conference on Artificial Intelligence in Education*. Berlin: Springer-Verlag.

Heffernan, N. T., & Croteau, E. A. (2004). Web-based evaluations showing differential learning for tutorial strategies employed by Ms. Lindquist tutor. In J. C. Lester, R. M. Vicari, & F. Parguaca (Eds.), *Intelligent Tutoring Systems: 7th International Conference, ITS 2004* (pp. 491-500). Berlin: Springer-Verlag.

Heffernan, N. T., & Koedinger, K. R. (1997). The composition effect in symbolizing: The role of symbol production vs. text comprehension. In M. G. Shafto & P. Langley (Eds.), *Proceedings of the Ninetheenth Annual Meeting of the Cognitive Science Society* (pp. 307-312). Mahwah, NJ: Erlbaum.

Heffernan, N. T., Koedinger, K. R., & Razzaq, L. (2008). Expanding the Model-Tracing Architecture: A 3rd generation intelligent tutor for algebra symbolization. *International Journal of Artificial Intelligence in Education, 18*, 153-178.

Hinsley, D. A., Hayes, J. R., & Simon, H. A. (1977). From words to equations: Meaning and representation in algebra word problems. In P. Carpenter & M. A. Just (Eds.), *Cognitive Processes in Comprehension*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Hoffer, T. B., Venkataram, L., Hedberg, E. C., & Shagle, S. (2007). *Final Report on the National Survey of Algebra Teachers for the National Math Panel*. Retrieved from Chicago, IL:

Hutchinson, N. L. (1993). Effects of cognitive strategy instruction on algebra problem solving of adolescents with learning disabilities. *Learning Disability Quarterly, 16*, 34-63.

Jitendra, A. K., Griffin, C. C., Haria, P., Leh, J., Adams, A., & Kaduvettoor, A. (2007). A comparison of single and multiple strategy instruction on third-grade students' mathematical problem solving. *Journal of Educational Psychology, 99*(1), 115-127.

Jitendra, A. K., Harwell, M. R., Dupuis, D. N., Karl, S. R., Lein, A. E., Simonson, G., & Slater, S. C. (2015). Effects of a research-based intervention to improve seventh-grade students' proportional problem solving: A cluster randomized trial. *Journal of Educational Psychology, 107*(4), 1019-1034.

Jitendra, A. K., Star, J. R., Dupuis, D. N., & Rodiguez, M. C. (2013). Effectiveness of schema-based instruction for improving seventh-grades students' proportional reasoning: A randomized experiment. *Journal of Research on Educational Effectiveness, 6*(2), 114-136.

Jitendra, A. K., Star, J. R., Rodriguez, M., Lindell, M., & Someki, F. (2011). Improving students' proportional thinking using schema-based instruction. *Learning and Instruction, 21*, 731-745.

Jitendra, A. K., Star, J. R., Starosta, K., Leh, J., Sood, S., Caskie, G., . . . Mack, T. r. (2009). Improving seventh grade students' learning of ratio and proportion: The role of schema-based instruction. *Contemporary Educational Psychology, 34*, 250-264.

Katz, S., Allbritton, D., & Connelly, J. (2003). Going beyond the problem given: How human tutors use post-solution discussions to support transfer. *International Journal of Artificial Intelligence in Education, 13*, 79-116.

Katz, S., Connelly, J., & Wilson, C. (2007). Out of the lab and into the classroom: An evaluation of reflective dialogue in Andes. In R. Luckin & K. R. Koedinger (Eds.), *Proceedings of AI in Education, 2007* (pp. 425-432). Amsterdam, Netherlands: IOS Press.

Kintsch, W., & Greeno, J. G. (1985). Understanding and solving word arithmetic problems. *Psychological Review, 92*, 109-129.

Koedinger, K. R., Alibali, M. W., & Nathan, M. J. (2008). Trade-offs between grounded and abstract representations: Evidence from algebra problem solving. *Cognitive Science, 32*, 366-397.

Koedinger, K. R., & Anderson, J. R. (1998). Illustrating principled design: The early evolution of a cognitive tutor for algebra symbolization. *Interactive Learning Environments, 5*, 161-180.

Koedinger, K. R., Corbett, A., & Perfetti, C. (2012). The Knowledge-Learning-Instruction (KLI) framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science, 36*(5), 757-798.

Leelawong, K., & Biswas, G. (2008). Designing learning by teaching agents: The Betty's Brain system. *International Journal of Artificial Intelligence and Education, 18*(3), 181-208.

Löhner, S., Van Joolingen, W. R., & Savelsbergh, E. R. (2003). The effect of external representation on constructing computer models of complex phenomena. *Instructional Science, 31*, 395-418.

Löhner, S., Van Joolingen, W. R., Savelsbergh, E. R., & Van Hout-Wolters, B. (2005). Students' reasoning during modeling in an inquiry learning environment. *Computers in Human Behavior, 21*, 441-461.

Looi, C.-K., & Tan, B. T. (1996). *WORDMATH: A computer-based environment for learning word problem solving*. Paper presented at the Conmputer Aided Learning and Instruction in Science and Engineering, San Sebastian, Spain.

Looi, C.-K., & Tan, B. T. (1998). A cognitive apprenticeship-based environment for learning word problem solving. *Journal of Computers in Mathematics abnd Science Teaching, 17*(4).

Marshall, S. P. (1995). *Schemas in Problem Solving*. Cambridge, UK: Cambridge University Press.

McArthur, D., Lewis, M., Ormseth, T., Robyn, A., Stasz, C., & Voreck, D. (1989). *Algebraic thinking tools: Support for modeling situations and solving problems in Kids' World*. Retrieved from Santa Monica, CA:

Metcalf, S. J. (1999). *The design of guided learning-adaptable scaffolding in interactive learning environments.* (Ph. D.), University of Michigan, Ann Arbor, MI.

Metcalf, S. J., Krajcik, J., & Soloway, E. (2000). Model-It: A design retrospective. In M. J. Jacobson & R. B. Kozma (Eds.), *Innovations in Science and Mathematics Education: Advanced Designs for Technologies of Learning* (pp. 77-115).

Munez, D., Orrantia, J., & Rosales, J. (2013). The effect of external representations on compare word problems: Supporting mental model construction. *Journal of Experimental Education, 81*(3), 337-355. doi:10.1080/00220973.2012.715095

Nathan, M. J., Kintsch, W., & Young, E. (1992). A theory of algebra-word-problem comprehension and its implications for the design of learning enviroments. *Cognition and Instruction, 9*(4), 329-389.

NGA & CCSSO. (2011). Common Core State Standards for Mathematics In: Downloaded from www.corestandards.org on October 31, 2011.

NGSS. (2013). *Next Generation Science Standards: For States, By States*: The National Academies.

Pareto, L., Arvemo, T., Dahl, Y., Haake, M., & Gulz, A. (2011). A teachable-agent arithmetic game's effects on mathematics understanding, attitude and self-efficacy. In G. Biswas & S. Bull (Eds.), *Proceedings of Artificial Intelligence in Education* (pp. 247-255). Berlin: Springer.

Pauli, C., & Reusser, K. (1997). *Supporting collaborative problem solving: Supporting collaboration and supporting problem solving*. Paper presented at the Proceedings of Swiss Workshop on Collaborative and Distributed Systems.

Quinn, J., & Alessi, S. M. (1994). The effects of simulation complexity and hypothesis-generation strategy on learning. *Journal of Research in Computing in Education, 27*(1), 75-92.
Ramachandran, S. (2003). A meta-cognitive computer-based tutor for high-school algebra. In D. Lassner & C. McNaught (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2003* (pp. 911-914). Chesapeake, VA: AACE.
Reed, S. K. (1998). *Word Problems: Research and Curriculum Reform*. New York: Routledge.
Reif, F., & Scott, L. A. (1999). Teaching scientific thinking skills: Students and computers coaching each other. *American Journal of Physics, 67*(9), 819-831.
Reimann, P. (2011). Design-based research. In L. Markauskaite, P. Freebody, & J. Irwin (Eds.), *Methodological Choice and Design: Scholarship, Policy and Practice in Social and Educational Research* (pp. 37-50). Berline: Springer.
Renkl, A., Stark, R., Gruber, H., & Mandl, H. (1998). Learning from worked-out examples: The effects of example variability and elicited self-explanations. *Contemporary Educational Psychology, 23*, 90-108.
Reusser, K. (1993). Tutoring systems and pedagogical theory: Representational tools for understanding, planning and reflection in problem solving. In S. P. Lajoie & S. J. Derry (Eds.), *Computers as Cognitive Tools* (pp. 143-178). Hillsdale, NJ: Lawrence Erlbaum Associates.
Reusser, K. (1996). From cognitive modeling to the design of pedagogical tools. In S. Vosniadou, E. De Corte, R. Glaser, & H. Mandl (Eds.), *International Perspectives on the Design of Technology-Supported Learning Environmetns*. Mahwah, NJ: Erlbaum.
Riley, M. S., & Greeno, J. G. (1988). Developmental analysis of understanding language about quantities and of solving problems. *Cognition and Instruction, 5*(1), 49-101.
Schwartz, D. L., Chase, C., Chin, D. B., Oppezzo, M., Kwong, H., Okita, S. Y., . . . Wagster, J. (2009). Interactive metacognition: Monitoring and regulating a teachable agent. In D. J. Hacker, J. Dunlosky, & A. C. Graesser (Eds.), *Handbook of Metacognition in Education.* (pp. 340-358). New York: Taylor & Francis.
Segedy, J. R., Kinnebrew, J. S., & Biswas, G. (2012). *Supporting student learning using converstational agents in a teachable agent environment*. Paper presented at the Proceedings of the 10th International Conference of the Learning Sciences, Sydney, Australia.
Shih, B., Koedinger, K. R., & Scheines, R. (2008). A response time model for bottom-out hints as worked examples. In C. Romero, S. Ventura, M. Pechenizkiy, & R. S. J. d. Baker (Eds.), *Handbook of Educational Data Mining* (pp. 201-211). Boca Raton, FL: Taylor & Francis.
Simsek, E., Xenidou-Deryou, I., Karadeniz, I., & Jones, I. (2019). The conception of substituion of the equals sign plays a unique role in students' algebra performance. *Journal of Numerical Cognition, 5*, 24-37.
Swaak, J., van Joolingen, W. R., & de Jong, T. (1998). Supporting simulation-based learning; The effects of model progression and assignments on definition and intuitive knowledge. *Learning and Instruction, 8*(3), 235-252.
van Joolingen, W. R., De Jong, T., Lazonder, A., Savelsbergh, E. R., & Manlove, S. (2005). Co-Lab: Research and development of an online learning environment for collaborative scientific discovery learning. *Computers in Human Behavior, 21*, 671-688.
VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence and Education, 16*, 227-265.
VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems and other tutoring systems. *Educational Psychologist, 46*(4), 197-221.
VanLehn, K. (2013). Model construction as a learning activity: A design space and review. *Interactive Learning Environments, 21*(4), 371-413.
Vanlehn, K., & Chi, M. (2012). Adaptive expertise as acceleration of future learning: A case study. In P. J. Durlach & A. Lesgold (Eds.), *Adaptive Technologies for Training and Education*. Cambridge: Cambridge University Press.
VanLehn, K., Chung, G., Grover, S., Madni, A., & Wetzel, J. (2016). Learning science by constructing models: Can Dragoon increase learning without increasing the time required? *International Journal of Artificial Intelligence in Education, 26*(4), 1033-1068.
VanLehn, K., Graesser, A. C., Jackson, G. T., Jordan, P., Olney, A., & Rose, C. P. (2007). When are tutorial dialogues more effective than reading? *Cognitive Science, 31*(1), 3-62.
VanLehn, K., Lynch, C., Schultz, K., Shapiro, J. A., Shelby, R. H., Taylor, L., . . . Wintersgill, M. C. (2005). The Andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence and Education, 15*(3), 147-204.

VanLehn, K., Wetzel, J., Grover, S., & van de Sande, B. (2017). Learning how to construct models of dynamic systems: An initial evaluation of the Dragoon intelligent tutoring system. . *IEEE Transactions on Learning Technologies, 10*(2), 154-167.

Wetzel, J., VanLehn, K., Chaudhari, P., Desai, A., Feng, J., Grover, S., . . . van de Sande, B. (2016). The design and development of the Dragoon intelligent tutoring system for model construction: Lessons learned. . *Interactive Learning Environments, 25*(3), 361-381. doi:10.1080/10494820.2015.1131167

White, B. Y. (1984). Designing computer games to help physics students understand Newton's Laws of Motion. *Cognition and Instruction, 1*(1), 69-108.

White, B. Y. (1993). ThinkerTools: Causal models, conceptual change and science education. *Cognition and Instruction, 10*(1), 1-100.

White, B. Y., & Frederiksen, J. R. (1990). Causal model progressions as a foundation for intelligent learning environments. *Artificial Intelligence, 42*, 99-157.

Willis, G. B., & Fuson, K. C. (1988). Teaching children to use schematic drawings to solve addition and subtraction word problems. *Journal of Educational Psychology, 80*(2), 192-201.

Xin, Y. P., Jitendra, A. K., & Deatline-Buchman, A. (2005). Effects of mathematical word problem-solving instruction on middle school students with learning problems. *The Journal of Special Education, 39*(3), 181-192.

Xin, Y. P., Zhang, D., Park, J. Y., Tom, K., Whipple, A., & Si, L. (2001). A comparison of two mathematics problem-solving strategies: Facilitate algebra-readiness. *The Journal of Educational Research, 104*(6), 381-395.

Zhang, L., Vanlehn, K., Girard, S., Burleson, W., Chavez-Echeagaray, M.-E., Gonzalez-Sanchez, J., & Hidalgo Pontet, Y. (2014). Evaluation of a meta-tutor for constructing models of dynamic systems. *Computers & Education, 75*, 196-217.