Teaching Underachieving Algebra Students to Construct Models Using a Simple Intelligent Tutoring System

Kurt VanLehn(✉), Fabio Milner, Chandrani Banerjee, and Jon Wetzel

Arizona State University, Tempe, AZ, USA
{kurt.vanLehn,milner,cbanerj1,jwetzel4}@asu.edu

Abstract. An algebraic model uses a set of algebraic equations to describe a situation. Constructing such models is a fundamental skill, but many students still lack the skill, even after taking several algebra courses in high school and college. For underachieving college students, we developed a tutoring system that taught students to decompose the to-be-modelled situation into schema applications, where a schema represents a simple relationship such as distance-rate-time or part-whole. However, when a model consists of multiple schema applications, it needs some connection among them, usually represented by letting the same variable appear in the slots of two or more schemas. Students in our studies seemed to have more trouble identifying connections among schemas than identifying the schema applications themselves. This paper describes a newly designed tutoring system that emphasizes such connections. An evaluation was conducted using a regression discontinuity design. It produced a marginally reliable positive effect of moderate size (d = 0.4).

Keywords: Intelligent tutoring system · Algebraic model construction · Algebra story problem solving · Algebra word problem solving

1 The Research Problem and Prior Work on It

Constructing models is a fundamental and important skill. According to the Next Generation Science Standards [1], "developing and using models" is one of 8 key scientific practices. According to the Common Core State Standards for Mathematics (CCSSM) [2], "modeling with mathematics" is one of its 8 key mathematical practices.

Students are introduced to model construction with arithmetic story problems in primary school, and then algebraic story problems in secondary school. Both are notoriously difficult. Many methods for teaching model construction have been investigated [see 3 for review].

Several researchers have applied Kintch's theory of text comprehension to model construction [4–12]. The theory posits that students construct equations by matching schemas against their understanding of the story. Each match of a schema fills slots of the schema and produces an equation. Nathan et al. [6] observed that some relationships were obvious to students and some were not. For example, given this story:

I. Roll et al. (Eds.): AIED 2021, LNAI 12749, pp. 367–371, 2021.
https://doi.org/10.1007/978-3-030-78270-2_65

Six seconds 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?

the equations below illustrate a model that conforms to the theory.

- *Dplane* = *537* * *Tplane*; obvious application of the Motion schema
- *Dmissile* = *740* * *Tmissile*; obvious application of the Motion schema
- *Dplane* = *Dmissile*; nonobvious application of the Overtake schema
- *Tplane* = *Tmissile* + *6*; nonobvious application of the Comparison schema

In a series of design-based research and quantitative studies, we converged on an instructional design that taught schemas explicitly using an example-based intelligent tutoring system [13]. The results were positive but not statistically reliable.

In order to understand the remaining impediments to learning, we tutored students individually. Students seemed to have more trouble identifying connections among schemas than identifying the schema applications themselves. We redesigned the instruction to replace the traditional concept of a variable denoting a quantity with variables as connections between slots. The next section describes the tutoring system.

2 The OMRaaT Tutoring System

Figure 1 shows a solved problem in the new tutoring system, which is named OMRaaT: an acronym for One Mathematical Relationship at a Time. Each row is a schema application. The boxes are slots. When students select the name of a schema, a new row is added to the table with empty slots labelled by the text above them. The first few slots describe the schema application. The Motion schema (first two rows of Fig. 1) has 4 description slots; the Equality schema (third row) has 2; the Addition schema (last row) has 2. Students fill a description slot by selecting from a menu including all possible description slot fillers.

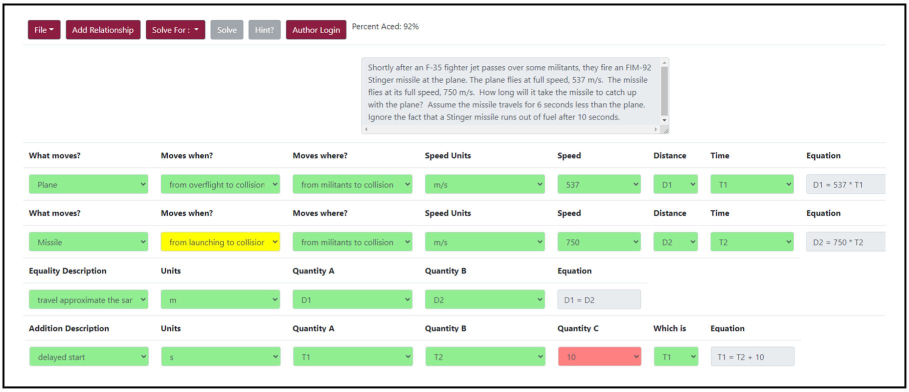

Fig. 1. A solved problem in OMRaaT

When the student finishes filling all the description slots of a schema application, the slots turn red (incorrect) or green (correct). On the third incorrect attempt, the slot turns yellow and shows the correct entry. The yellow coloring and the delayed feedback are intended to discourage guessing. Also, the percentage of slots filled correctly on the first attempt is displayed at the top of the window (e.g., "Percent Aced: 92%").

After all the description slots of a schema application have been filled correctly, the student fills in the remaining "quantity" slots. For the Motion schema, there are 3 quantity slots, for distance, rate and time. To fill a quantity slot of the Motion schema or other obvious schemas, students select from a menu that has numbers mentioned in the story (e.g., 537, 750 and 6 for the example above) and "Unknown." If they select Unknown, the system invents a variable name that is unique to the slot. Thus, the variables denote slots. When all the quantity slots have been filled, the student gets red/green feedback.

When students fill the two quantity slots of an Equality schema application, they select from a menu that has only the variables defined in the obvious schema applications. When they fill the quantity slots in an Addition schema application, they select from a menu with Unknown, all the variables of obvious schema applications and all the numbers in the story. The Addition schema has students first enter 3 quantities involved in an addition relationship, then select one of them to be the sum.

When students have finished correctly entering all the schema applications required, they click the "Solve for:" button to select one of the variables, then click the Solve button, which pops up a window with the numerical value of the solved-for variable.

The overall instruction was implemented as a module of 60 pages in the university's LMS, Canvas. Most pages were OMRaaT problems. A few pages were text with examples. A few more pages were problems to be done on the assessment system, called the Solver. To solve a problem on the Solver, the student types in equations using only numbers that appear in the problem statement. When the student presses the Solve button, it solves the equations or displays an error message. Every Solver-problem page was followed by a page with a video of the instructor solving the same Solver problem using OMRaaT-style reasoning.

3 An Evaluation

To evaluate the OMRaaT module, we used a regression discontinuity design in the context of an undergraduate class on modeling. The class historically has many mathematically challenged students. Students above the cutoff on the Solver-based pretest (the no-treatment group) were prevented from taking the OMRaaT modules. Students below the cutoff (the treatment group) were required to take the module. They could be considered underachieving students with regards to algebraic modeling skill.

To be fair to the students in both groups, algebraic modeling was taught twice bracketed by tests. The sequence was: (1) pretest, (2) OMRaaT module for the treatment group only, (3) mid-test, (4) Solver-based instruction on algebraic modelling for all students, and (5) post-test. Students' scores on the mid-test and post-test counted towards their grade, whereas the score on the pre-test was only used for placement in the treatment or no-treatment condition.

Of the 53 students who consented to have their data used for this evaluation and took both the pre-test and mid-test, 29 were in the treatment group and 23 were in the no-treatment group. One student whose pre-test score was below the cutoff did only 4 of the assigned OMRaaT problems, and thus was excluded from the treatment group. All the other treatment students did most of the problems.

Figure 2 shows the regression plot for the two groups. The solid line is the diagonal, where mid-test scores equals pre-test scores. The good news is that all but one of the treatment students gained: their mid-test scores were larger than their pre-test scores, often by large amounts. The majority of the no-treatment students also gained, which could be due to a test-retest effect or gaining familiarity with the Solver. However, a significant proportion (approximately one-third) of the no-treatment students actually decreased their scores from pre-test to mid-test. The normalized gain scores of the treatment group were higher than those of the no-treatment group, but not reliably so ($p = 0.08$, $d = 0.42$).

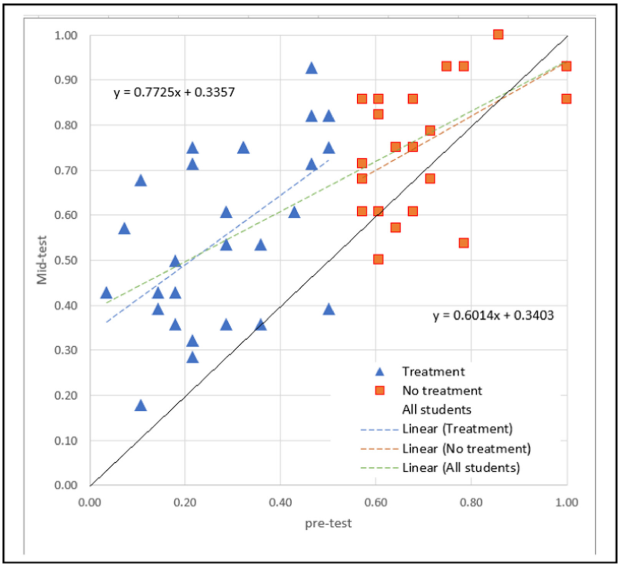

Fig. 2. Results of the OMRaaT module evaluation

However, the success of a regression discontinuity design hinges on whether the regression lines of the two groups (shown in blue and red) fit the data better than a regression line for the union of the two groups (shown in green). The OMRaaT module raised the regression line of the treatment group by about 0.09 above the no-treatment group. As the standard deviation of the post-test scores was 0.21, an 0.09 increase corresponds to an effect size of 0.41. However, the double regression line model was not reliability different from the single regression line model ($p = 0.27$). Our interpretation is that there may be a positive benefit, but it is too small to show up reliably given the large scatter in the data and the small number of data points.

The bottom line is that there appears to be a moderately large positive effect ($d = 0.4$), but its existence is doubtful due to the large variance in scores and the small sample. The fact that the OMRaaT module improved the scores of underachieving students is remarkable and welcome, but more data are needed to be sure that the positive apparent benefit actually exists.

References

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

2. Common Core State Standards for Mathematics (2011)
3. VanLehn, K.: Model construction as a learning activity: a design space and review. Interact. Learn. Environ. **21**(4), 371–413 (2013)
4. Kintsch, W., Greeno, J.G.: Understanding and solving word arithmetic problems. Psychol. Rev. **92**, 109–129 (1985)
5. Cummins, D.D., Kintsch, W., Reusser, K., Weimer, R.: The role of understanding in solving word problems. Cogn. Psychol. **20**, 405–438 (1988)
6. Nathan, M.J., Kintsch, W., Young, E.: A theory of algebra-word-problem comprehension and its implications for the design of learning environments. Cogn. Instr. **9**(4), 329–389 (1992)
7. Fuchs, L., Fuchs, D., Seethaler, P.M., Barnes, M.A.: Addressing the role of working memory in mathematical word-problem solving when designing intervention for struggling learners. ZDM Math. Educ. **52**(1), 87–96 (2019). https://doi.org/10.1007/s11858-019-01070-8
8. Riley, M.S., Greeno, J.G.: Developmental analysis of understanding language about quantities and of solving problems. Cogn. Instr. **5**(1), 49–101 (1988)
9. Marshall, S.P.: Schemas in Problem Solving. Cambridge University Press, Cambridge (1995)
10. Fuchs, L.S., et al.: The effects of schema-broadening instruction on second grader's word-problem performance and their ability to represent word problems with algebraic equations: a randomized control study. Elem. Sch. J. **110**(4), 440–463 (2010)
11. Jitendra, A.K., et al.: Effects of a research-based intervention to improve seventh-grade students' proportional problem solving: a cluster randomized trial. J. Educ. Psychol. **107**(4), 1019–1034 (2015)
12. Xin, Y.P., Jitendra, A.K., Deatline-Buchman, A.: Effects of mathematical word problem-solving instruction on middle school students with learning problems. J. Spec. Educ. **39**(3), 181–192 (2005)
13. Vanlehn, K., Banerjee, C., Milner, F., Wetzel, J.: Teaching algebraic model construction: a tutoring system, lessons learned and an evaluation. Int. J. Artif. Intell. Educ. **30**(3), 459–480 (2020)