A General Multi-method Approach to Data-Driven Redesign of Tutoring Systems

Yun Huang
yunhuanghci@cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

Nikki G. Lobczowski
nikkilob@cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

J. Elizabeth Richey
jelizabethrichey@cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

Elizabeth A. McLaughlin
mimim@cs.cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

Michael W. Asher
mwasher@wisc.edu
University of Wisconsin-Madison
Madison, WI, USA

Judith M. Harackiewicz
jmharack@wisc.edu
University of Wisconsin-Madison
Madison, WI, USA

Vincent Aleven
aleven@cs.cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

Kenneth R. Koedinger
koedinger@cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

ABSTRACT

Analytics of student learning data are increasingly important for continuous redesign and improvement of tutoring systems and courses. There is still a lack of general guidance on converting analytics into better system design, and on combining multiple methods to maximally improve a tutor. We present a multi-method approach to data-driven redesign of tutoring systems and its empirical evaluation. Our approach systematically combines existing and new learning analytics and instructional design methods. In particular, our methods involve identifying difficult skills and creating focused tasks for learning these difficult skills effectively following content redesign strategies derived from analytics. In our past work, we applied this approach to redesigning an algebraic modeling unit and found initial evidence of its effectiveness. In the current work, we extended this approach and applied it to redesigning two other tutor units in addition to a second iteration of redesigning the previously redesigned unit. We conducted a one-month classroom experiment with 129 high school students. Compared to the original tutor, the redesigned tutor led to significantly higher learning outcomes, with time mainly allocated to focused tasks rather than original full tasks. Moreover, it reduced over- and under-practice, yielded a more effective practice experience, and selected skills progressing from easier to harder to a greater degree. Our work provides empirical evidence of the effectiveness and generality of a multi-method approach to data-driven instructional redesign.

CCS CONCEPTS

• **Applied computing** → **E-learning**; **Computer-managed instruction**; *Interactive learning environments.*

LAK21, April 12–16, 2021, Irvine, CA, USA
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8935-8/21/04.
https://doi.org/10.1145/3448139.3448155

KEYWORDS

instructional design, learning design, learning engineering, data mining, adaptivity

ACM Reference Format:
Yun Huang, Nikki G. Lobczowski, J. Elizabeth Richey, Elizabeth A. McLaughlin, Michael W. Asher, Judith M. Harackiewicz, Vincent Aleven, and Kenneth R. Koedinger. 2021. A General Multi-method Approach to Data-Driven Redesign of Tutoring Systems. In *LAK21: 11th International Learning Analytics and Knowledge Conference (LAK21), April 12–16, 2021, Irvine, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3448139.3448155

1 THE NEED FOR A GENERAL DATA-DRIVEN REDESIGN APPROACH

In recent years, there have been a growing number of endeavors to apply data-driven methods to the continuous improvement in courses and educational technologies such as the Loop tool [6] and Course Signals [5]. The notion of data-driven redesign of a course or a system based on analytics of student learning data is referred to as *design-loop adaptivity* in the Adaptivity Grid framework [2]. The rationale behind design-loop adaptivity is that an earlier iteration of a design made by domain experts may be suboptimal due to expert blind spot [29] or resource constraints, and analytics of data from previous iterations provides an efficient and effective way to uncover design deficiencies and to improve the design. Design-loop adaptivity, the theme of our current work, emphasizes between-system-iteration changes adapted to the demands of the task domain or students' similarities (e.g., content redesign), rather than or in addition to within-system changes adapted to students' differences (e.g., adaptive task selection).

Tutoring systems have become an integral part of courses in many blended or online learning contexts, generating a large amount of student learning data. Learning analytics of such data are increasingly being used to improve instruction and enhance student learning. Some empirical studies show that data-driven redesign of tutoring systems (i.e., design-loop adaptivity) can lead to better student learning outcomes. For example, Mostafavi and Barnes [28] augmented a logic tutor by several data-driven components adapting to students' differences such as intelligent problem selection, and found that it allowed students to be exposed to more concepts. Liu and Koedinger [23] used data-driven cognitive model discoveries to redesign instruction of a geometry tutor and showed that the redesigned tutor led to significantly higher learning gains relative to a control condition.

Meanwhile, numerous learning analytics and data mining methods have been demonstrated to improve prediction accuracy on student performance, but most stop at better predictions without demonstrating whether and how these methods can improve student learning in situ [10, 15, 31]. As pointed out in [11], the learning analytics cycle is not complete (i.e., the loop is not closed) until analytics are used to drive interventions that have some effect on learners. Examining the few close-the-loop empirical studies, there remains a critical limitation: they focused on a single design aspect such as adapting to students' differences [28, 33], or applied a single analytics method [24, 32], lacking general, explicit guidance on converting learning analytics outcomes into better system design.

In this paper, we demonstrate an approach that combines new and existing learning analytics and instructional design methods to redesign tutoring systems, and we describe a classroom experiment evaluating its effectiveness and generality, extending our prior approach and evaluation [17]. Our work makes the following contributions:

- Provide general, explicit guidance on converting analytics into better system design and instructional design;
- Provide general, explicit guidance on combining multiple methods focusing on different aspects to maximally improve a tutor, adapting to both students' similarities (i.e., demands of the task domain) and differences;
- Provide empirical evidence for the effectiveness and generality of our approach.

Our approach can be considered general in two senses. First, it is intended to be applicable to systems with learn-by-doing activities, which are usually the main activities in Cognitive Tutors [4] or an integral part of other systems with diverse activities such as MOOCs [9]. Doing activities have an estimated 6x greater impact on total quiz and final exam than reading or video watching [21]. Thus, system design improvements based on analytics of doing activities has the potential to yield substantial improvements in student learning.

More specifically, our approach is intended to be applicable to systems that follow a knowledge component (KC) approach to instruction [1], i.e., design and organize activities and instructions according to an underlying KC model. A *KC model* decomposes domain knowledge into small units (which can be called KCs or skills) and specifies the requisite knowledge units for tasks or steps of tasks. Our notion of a KC or a skill refers to a small unit that is needed for a single step in a tutor problem (task), rather than a broad notion like a topic area. A KC approach to instruction can dramatically improve the effectiveness of instruction, even without run-time adaptivity (i.e., tutoring systems don't have to be "intelligent" in the sense of adaptive task selection). For example, in the redesign of a blended OLI-statistics course [24], they uncovered deficiencies in the KC model and revised course objectives and activities accordingly. The redesigned course led to better learning outcomes in half the time, compared to the original course. In addition, a KC approach to instruction for individualized mastery learning can further improve instruction [2, 12], where a tutor continues to provide practice of targeted KCs until a student reaches mastery of each targeted KC in the given topic area (i.e., the estimated knowledge probability of each KC reaches a predefined threshold), before moving on to a new topic area. This cognitive mastery decision is achieved by adaptive task selection based on a student model which dynamically tracks each student's knowledge per KC through a statistical model of learning. We summarize three general challenges of this KC-based design-loop adaptivity:

- To create an *accurate* KC model that explains students' performance and learning transfer well for the task domain;
- To design content that facilitates (e.g., scaffolds) the learning of hard KCs;
- To make sure the tutor effectively distributes practice time across KCs for individual students.

Our multi-method redesign approach is designed to address these three challenges. A simple example motivating our redesign approach is as follows. In our original algebra tutor, all steps that require writing an expression are labeled with the KC "write an expression" and there is no difference in scaffolding between tasks. Having such a single KC implies that 1-operator practice transfers to 2-operator practice, and the tutor considers a student having mastered the KC if he or she performs well on 1-operator steps. However, as revealed by analytics (described later), students performed significantly worse on 2-operator steps than on 1-operator steps, and error rates of 2-operator steps remained high while those of 1-operator steps steadily decreased throughout the practice. This suggests a need to split the original KC into two KCs covering 1-operator and 2-operator expressions separately (challenge #1), and to design tailored tasks to address difficulties in 2-operator expressions (challenge #2). Further, the tutor should assign tasks so that each individual can practice both 1-operator and 2-operator KCs to mastery (challenge #3). Although there is a sizable body of research addressing the first and third challenges through data mining methods, there is a dearth of research on deriving explicit content design strategies from analytics, and on combining methods to address all three changes. Here, we articulate the *Focused Practice Task Design* method where we derived explicit content redesign strategies from analytics for creating focused tasks that optimize *deliberate practice*, a well-established way to support learning by doing through repetition with feedback on well-tailored tasks [14]. We also demonstrate how we combine methods to address the three challenges to maximumly improve a tutor.

In our past work, we demonstrated a multi-method redesign approach [17] used to redesign an algebraic modeling unit and we obtained initial empirical evidence of its effectiveness. In the current work, we extended the approach by adding new goals and methods. We applied the approach to redesigning two new units in addition to a second iteration of redesigning the unit that we previously redesigned. We conducted a larger-scale, longer-span experiment with comprehensive analyses for understanding the processes leading to the overall results. We now describe our redesign approach and classroom experiment.

2 HOW TO USE DATA TO REDESIGN TUTORING SYSTEMS

We created an approach called *MADDRED* (Multi-method Approach to Data-Driven REDesign) that uses a principled combination of methods to redesign tutoring systems utilizing learning data collected from previous iterations (see Table 1).[1] It is driven by the main goal of identifying KCs that are demonstrably difficult for students to learn, and to revise content and task selection to optimize effective and efficient practice on them. To clarify the terminologies in the table, an *opportunity* refers to a student's first attempt at a step that requires a KC, and it can be correct or incorrect.

We applied this approach to redesigning three algebraic units in Mathtutor [3], an online tutor with comprehensive content for middle-school mathematics. It was designed based on best practices and prior instructional design research [19], but had not been data-tuned. Figure 1 shows an original problem in the 1/2-operator Modeling unit, the unit we redesigned previously and in the current work as a second iteration. Here, students learn to write 1-operator or 2-operator algebraic expressions (e.g., $5x+25$) to model real-world situations. Figure 2 shows an original problem in the 2-operator Explanation unit; the 3-operator Explanation unit has the same format but involves 3-operator expressions. These two Explanation units were the new units we redesigned in the current work. In these units, students learn to explain sub-expressions from equations modeling real-world situations. We utilized log data from 499 students with 53,596 transactions collected from the original tutor to conduct our redesign. We describe our new methods below.

Difficulty Factor Effect Analysis. We created this method for KC model refinement so that a KC model used as the basis for instruction captures task distinctions that are important for novices, because such distinctions can be initially ignored due to expert blind spot [29]. A *difficulty factor* (DF) refers to a property that makes some task steps more difficult than other comparable ones of a KC. For example, "involve a negative number or not" could be a DF for arithmetic KCs. DFs can help reveal hidden hard KCs where novices need deliberate practice and tailored scaffolding. Usually, DFs are hypothesized by domain experts and are used to refine a KC model by splitting original KCs into more fine-grained ones. An effective way to use DFs to refine KC models is the Learning Factor Analysis (LFA) method [7]. It automatically searches through a space of KC models created by incorporating hypothesized DFs, using a statistical model of learning, Additive Factors Model (AFM).

Our method may be viewed as an efficient simplification of LFA (for its "split" operator). First, we identified a broad set of potential DFs by automatically coding task step features hypothesized to impact difficulty (e.g., requiring parentheses or not). Second, for each targeted KC (i.e., using only data involving this KC), we ran a logistic regression model predicting students' performance with a full interaction among DFs, and examined the main and interaction effects of DFs on students' performance, controlling for student proficiencies and learning from prior opportunities of this KC. A KC was cumulatively split by a set of DFs when there was an interaction among them or by a DF when there was a main effect. Third, we validated the KC modifications by incrementally incorporating the modification of each KC. In each step, we obtained a new KC model by applying the split decision to the current KC while keeping other KCs unchanged, and compared the statistical fit on overall data using AFM, with that from a previous KC model. An example of our method is as follows. For the original KC "write an expression", we found a main effect of the DF "the number of operators", and an interaction between the DF "require parentheses or not" and the DF "is repeated in a problem or not". Thus, we split the original KC into six more fine-grained KCs (i.e., "1op first", "1op repeat", "2op no-par first", "2op no-par repeat", "2op par first", "2op par repeat"), resulting in a KC model with a better overall statistical fit.

Probability-Propagated Practice Estimation. We developed this method for estimating the number of practice opportunities needed for mastery and the amount of under- and over-practice for a student on a KC, to inform content redesign. Although our estimation technique builds on prior work [8, 22], our method differs from prior work in that it increases computational efficiency and it is used to inform content redesign. We examined mastery on each fine-grained KC identified in the previous step, following one widely adopted definition of mastery, namely, that the probability of a student knowing a KC is ≥ 0.95, based on performance on problem steps with the KC [12]. In our method, first, we fit parameters of a student model (e.g., Bayesian Knowledge Tracing (BKT) [12]) to the data. Second, we used the fitted model as the "ground truth" to estimate knowledge for each opportunity that occurred in the data, and extrapolated opportunities until the estimated knowledge reached the mastery threshold, if the estimated knowledge for the last observed opportunity did not reach the mastery threshold. We then obtained the estimates of opportunities needed for mastery for each student on each KC. Finally, we compared the estimates to actual opportunities to infer over- or under-practice: if the actual number of opportunities that occurred in the data is greater than the estimated number of opportunities required for mastery, then the student is inferred to have over-practiced this KC; if it is less, then the student is inferred to have under-practiced the KC.

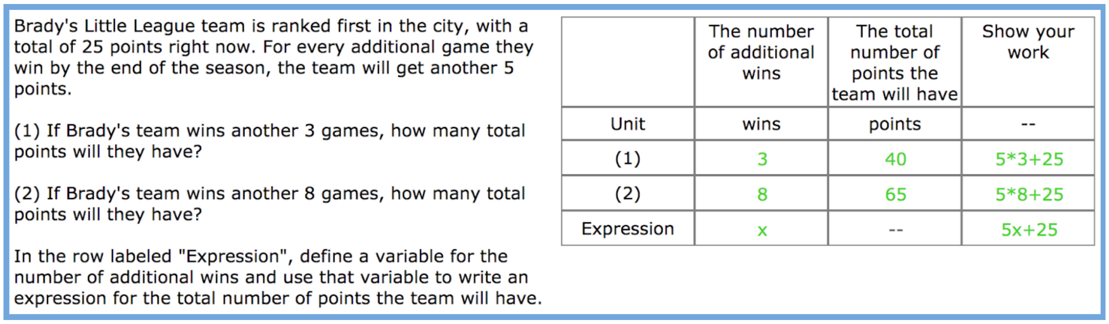
The challenge in this estimation is extrapolating student performance on unseen opportunities if the data lacks sufficient opportunities to reach mastery. Prior work did not consider or describe this process [8], or had relatively high computational time [22]. Here, instead of simulating a large number of sequences by propagating simulated outcomes as in [22], our method simulates one sequence by propagating the probability of succeeding (i.e., $P(C)$), and uses it as weights to update knowledge (i.e., $P(K)_{new}$=$P(C)P(K|C)_{new}$ $+P(W)P(K|W)_{new}$). The extrapolation of a KC-student sequence stops when the probability of the student knowing the KC reaches the mastery threshold (i.e., $P(K)_{new}\geq 0.95$), or the extrapolated opportunities reach a threshold (30). We chose 30 because it yielded similar estimations as the main existing method [22] in our dataset. We compared our method to the main existing method [22] in our dataset, and found that our method reached similar estimations with much higher computational efficiency. These estimations were used to inform content redesign in two ways: to inform the creation of focused tasks (explained below), and to inform the creation of content to ensure there would be sufficient practice opportunities for mastery (regardless of task selection).

Learning curve guided error analysis. In this analysis, we examined frequent errors on poorly performed practice opportunities according to learning curves, to better understand students'

[1] Our code is available on http://learnsphere.org and https://github.com/MADDRED.

Table 1: A general *m*ulti-method *a*pproach to *d*ata-*d*riven *red*esign of tutoring systems (MADDRED).

Goals	Methods
1 Refine the knowledge component (KC) model	
• Identify task factors that cause difficulties for learning KCs • Hypothesize and compare alternative KC models	Difficulty Factor Effect Analysis
2 Redesign content (adapting to students' similarities or the demands of the task domain)	
• Estimate opportunities to mastery, amount of under- and over-practice • Identify common errors and difficulties for hard KCs • Create focused tasks that target hard KCs • Add more content to ensure a sufficiency of content for mastery	Probability-Propagated Practice Estimation Learning curve guided error analysis Focused Practice Task Design Probability-Propagated Practice Estimation
3 Optimize individualized learning (adapting to students' differences)	
• Optimize student model parameters (based on a KC model) • Optimize adaptive task selection (based on a student model)	Data-tuning parameters [8], adaptive tutoring simulation Adaptive tutoring simulation



Brady's Little League team is ranked first in the city, with a total of 25 points right now. For every additional game they win by the end of the season, the team will get another 5 points.

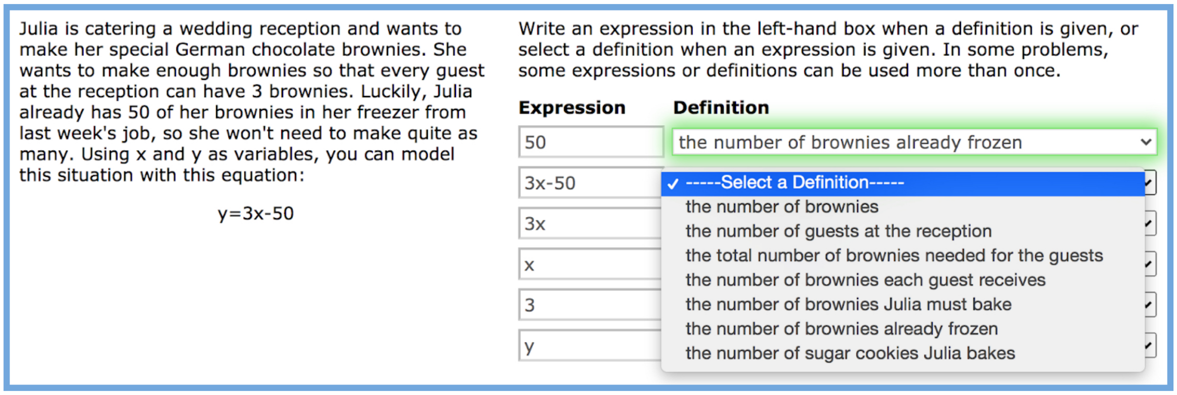
(1) If Brady's team wins another 3 games, how many total points will they have?

(2) If Brady's team wins another 8 games, how many total points will they have?

In the row labeled "Expression", define a variable for the number of additional wins and use that variable to write an expression for the total number of points the team will have.

	The number of additional wins	The total number of points the team will have	Show your work
Unit	wins	points	--
(1)	3	40	5*3+25
(2)	8	65	5*8+25
Expression	x	--	5x+25

Figure 1: A full task (problem) in the original Modeling unit (with cells filled in correctly and the toolbar excluded).



Julia is catering a wedding reception and wants to make her special German chocolate brownies. She wants to make enough brownies so that every guest at the reception can have 3 brownies. Luckily, Julia already has 50 of her brownies in her freezer from last week's job, so she won't need to make quite as many. Using x and y as variables, you can model this situation with this equation:

y=3x-50

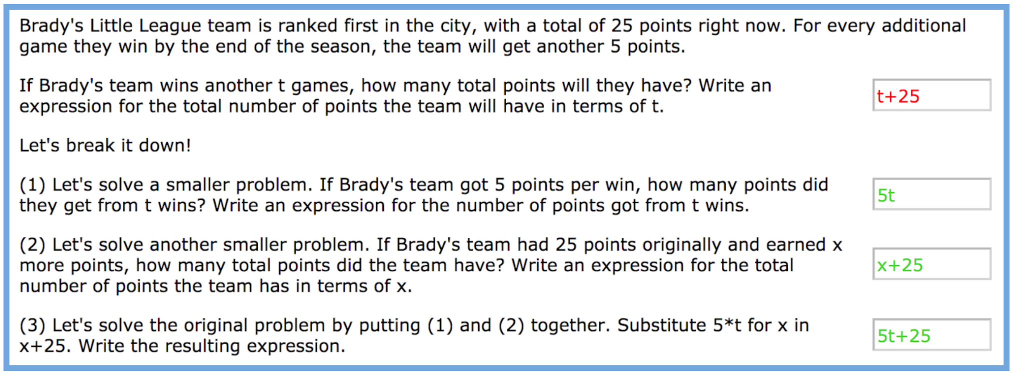
Write an expression in the left-hand box when a definition is given, or select a definition when an expression is given. In some problems, some expressions or definitions can be used more than once.

Expression	Definition
50	the number of brownies already frozen
3x-50	-----Select a Definition-----
3x	
x	
3	
y	

✓ -----Select a Definition-----
the number of brownies
the number of guests at the reception
the total number of brownies needed for the guests
the number of brownies each guest receives
the number of brownies Julia must bake
the number of brownies already frozen
the number of sugar cookies Julia bakes

Figure 2: A full task (problem) in the original 2-operator Explanation unit (with toolbar excluded).

difficulties on hard KCs for informing content redesign. A *learning curve* depicts the error rates (averaged over students) over successive practice opportunities for a KC or aggregated over KCs. Learning curves have been used to identify difficulty factors to refine KC models [32]. Here, we used learning curves to identify opportunities with high error rates (≥0.75) and examined the most frequent errors on such opportunities for each refined KC obtained before, utilizing the error report analytics tool from DataShop [20]. Such drill-down error analysis was used to inform the creation of focused tasks (explained below). For example, on the Explanation units, we identified a type of error, selecting an ambiguous description (e.g., "the number of brownies" vs. "the number of brownies Julia must bake" for the expression 3*x*-50 in Figure 2), that was not specific to any expressions (KCs). We thus created focused tasks with fewer KCs, providing feedback explaining this error, to address such common errors early on in simpler tasks.



Brady's Little League team is ranked first in the city, with a total of 25 points right now. For every additional game they win by the end of the season, the team will get another 5 points.

If Brady's team wins another t games, how many total points will they have? Write an expression for the total number of points the team will have in terms of t. t+25

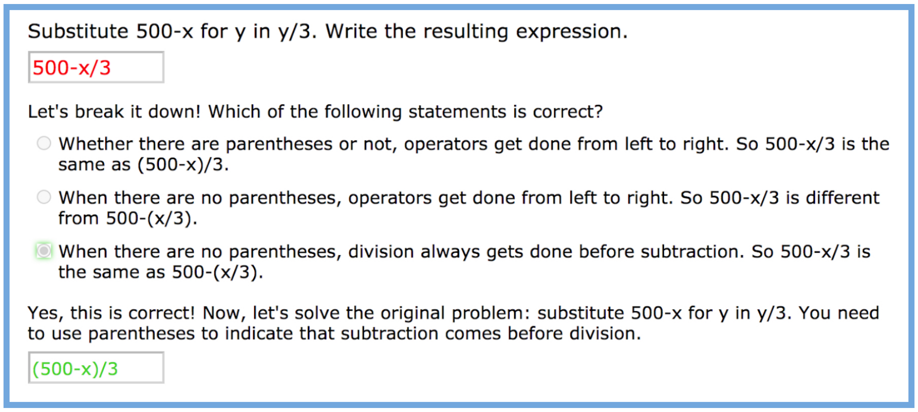
Let's break it down!

(1) Let's solve a smaller problem. If Brady's team got 5 points per win, how many points did they get from t wins? Write an expression for the number of points got from t wins. 5t

(2) Let's solve another smaller problem. If Brady's team had 25 points originally and earned x more points, how many total points did the team have? Write an expression for the total number of points the team has in terms of x. x+25

(3) Let's solve the original problem by putting (1) and (2) together. Substitute 5*t for x in x+25. Write the resulting expression. 5t+25

(a) A focused whole task here targets a hard KC without upfront fixed scaffolding steps training students to "putting it all together", in the same context as the original full task (Figure 1). If students fail on the whole task response (the first text field), composition scaffolding appears and isolates individual KCs (steps 1-3) including a hidden hard KC (e.g., expression embedding in step 3).



Substitute 500-x for y in y/3. Write the resulting expression.

500-x/3

Let's break it down! Which of the following statements is correct?

○ Whether there are parentheses or not, operators get done from left to right. So 500-x/3 is the same as (500-x)/3.

○ When there are no parentheses, operators get done from left to right. So 500-x/3 is different from 500-(x/3).

● When there are no parentheses, division always gets done before subtraction. So 500-x/3 is the same as 500-(x/3).

Yes, this is correct! Now, let's solve the original problem: substitute 500-x for y in y/3. You need to use parentheses to indicate that subtraction comes before division.

(500-x)/3

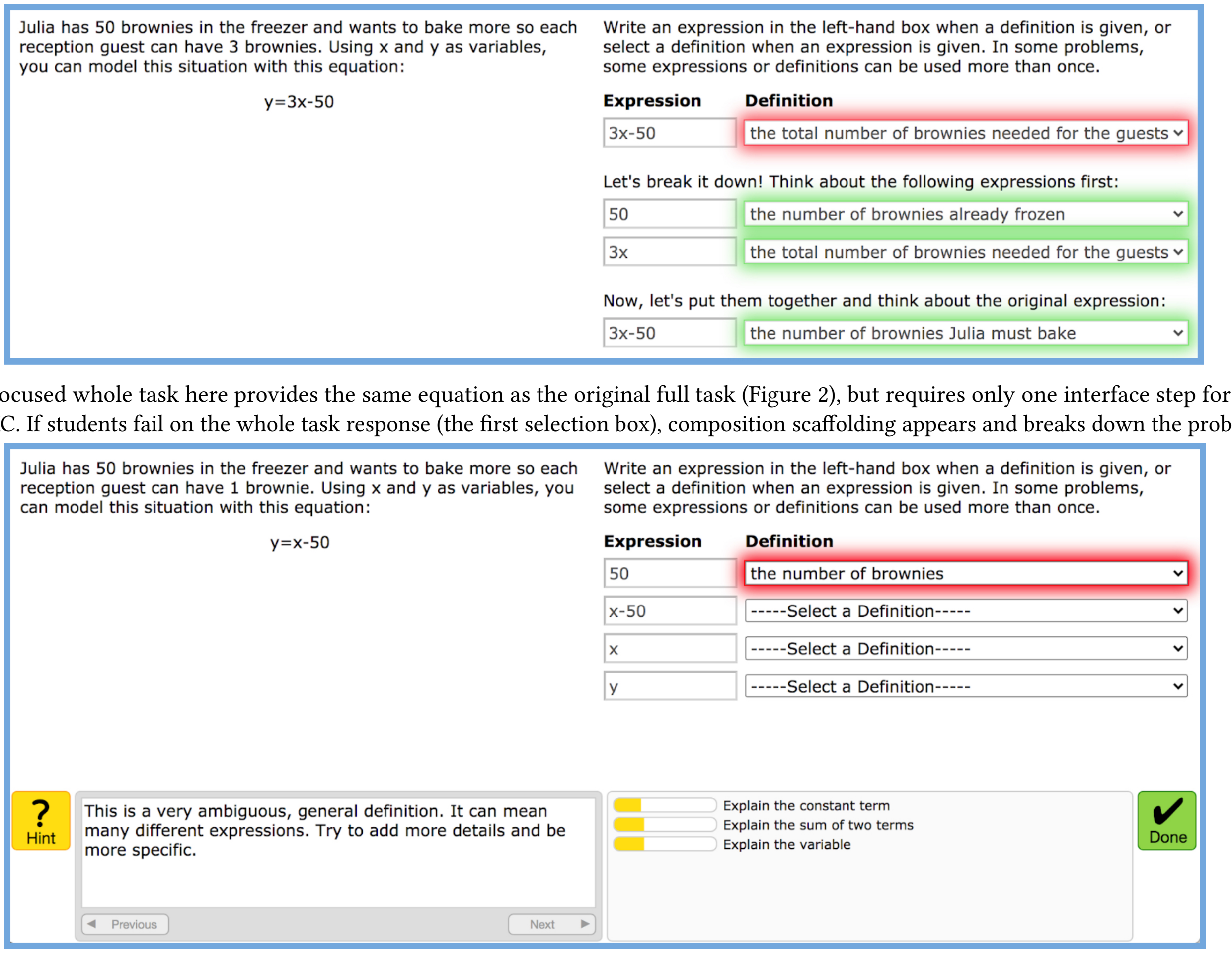
(b) A focused part task here targets a hidden hard KC that integrates other parts (e.g., integrates two 1-operator expressions into one 2-operator expression), in a smaller application context without requiring other KCs (e.g., story comprehension). If students commit specific errors (e.g., missing parentheses), scaffolding appears and shows a multiple-choice question for enhancing understanding.

Figure 3: Focused tasks in the redesigned Modeling unit to target hard KCs.

Focused Practice Task Design. This is a novel data-driven instructional design method where we created focused tasks that target hard KCs, following content redesign strategies derived from analytics of opportunities to mastery and of errors based on an accurate KC model, informed by prior cognitive and instructional design research. We created two kinds of focused tasks, namely, focused whole tasks and focused part tasks, which focus students' effort and attention for learning hard KCs in two different ways.

Focused whole tasks target a hard KC without upfront fixed scaffolding steps to perform in a larger application context. A *larger application context* means the problem statement of the original full task is mostly retained. As shown in Figure 3a and 4a, students are asked to directly enter the final answer, without being required to enter answers for easier, scaffolding steps in the interface. Unlike *typical* whole (full) tasks in Cognitive Tutors [4] (e.g., Figure 1, 2) where the interface has multiple interface steps corresponding with the multiple KCs needed in a whole task, in *focused* whole tasks this scaffolding is eliminated. The focus is on "putting it all together", demonstrating that students have acquired conditions on the target hard KC that will generalize outside the context of the scaffolding in typical whole tasks.

Focused part tasks target the hard KC(s) in isolation in a smaller application context where other KCs are eliminated in both interface steps and mental processes. Here, a *smaller application context* means the problem statement defines a simpler task than the original full or focused whole task. Focused part tasks facilitate the focus of student attention on the particular challenges required in learning and executing the target hard KC(s). Unlike *typical* part tasks designed to isolate each step including steps for easier KCs keeping the original task context, *focused* part tasks are often newly invented tasks with smaller contexts that isolate the challenging cognitive processing (KCs) that integrates other parts. For example, Figure 3b shows a focused part task that isolates the hidden hard KC that integrates two 1-operator expressions into one 2-operator expression (i.e., expression embedding), in a context without a cover story. Note that this step was implicit in the original full task (Figure 1), but was made explicit in focused tasks (Figure 3) thanks to



Julia has 50 brownies in the freezer and wants to bake more so each reception guest can have 3 brownies. Using x and y as variables, you can model this situation with this equation:

y=3x-50

Write an expression in the left-hand box when a definition is given, or select a definition when an expression is given. In some problems, some expressions or definitions can be used more than once.

Expression	Definition
3x-50	the total number of brownies needed for the guests

Let's break it down! Think about the following expressions first:

50	the number of brownies already frozen
3x	the total number of brownies needed for the guests

Now, let's put them together and think about the original expression:

3x-50	the number of brownies Julia must bake

(a) A focused whole task here provides the same equation as the original full task (Figure 2), but requires only one interface step for one hard KC. If students fail on the whole task response (the first selection box), composition scaffolding appears and breaks down the problem.

Julia has 50 brownies in the freezer and wants to bake more so each reception guest can have 1 brownie. Using x and y as variables, you can model this situation with this equation:

y=x-50

Write an expression in the left-hand box when a definition is given, or select a definition when an expression is given. In some problems, some expressions or definitions can be used more than once.

Expression	Definition
50	the number of brownies
x-50	-----Select a Definition-----
x	-----Select a Definition-----
y	-----Select a Definition-----

This is a very ambiguous, general definition. It can mean many different expressions. Try to add more details and be more specific.

(b) A focused part task here provides a simpler equation than the original full task (Figure 2), without requiring other hard KCs (e.g., explain $3x$). Easier KCs (e.g., explain 50) are retained for addressing common errors (e.g., selecting an ambiguous description) not specific to hard KCs, so that students could address them early on in such simpler tasks. Immediate feedback explaining common errors are provided.

Figure 4: Focused tasks in the redesigned 2-operator Explanation unit to target hard KCs.

our KC-based task design. Moreover, in both focused tasks, tailored scaffolding appears if and only if the initial answer is incorrect.

To create focused tasks, we derived three design strategies from analytics of opportunities to mastery and of errors based on an accurate KC model, informed by prior research (Table 2). Our strategies align with the goals pointed out by Moreno and Mayer [27] for addressing cognitive processing demands during learning, in that our focused tasks facilitate the focus of student attention on the particular challenges required in learning (generative processing) and executing (essential processing) the target hard KC(s), reducing unnecessary processing (extraneous processing) of easier KCs or other hard KCs. We explain our redesign strategies as follows.

Reduce over-practice on easier KCs and under-practice on hard KCs. According to our estimation, many students over-practiced easier KCs and under-practiced hard KCs in the original tutor. Our focused tasks target the hard KC(s) without requiring fixed steps of untargeted KCs (i.e., easier KCs or other hard KCs), and provide dynamic scaffolding. In this way, students can reduce inefficient use of their time on KCs too easy or on too many hard KCs, so that they can better use their time to learn each hard KC, and reduce over-practice on easier KCs and under-practice on hard KCs.

Provide effective scaffolding for hard KCs informed by prior research. Our method estimated that too many opportunities would be needed to master hard KCs in the original Modeling unit, suggesting that the original scaffolding may not be effective enough. Prior cognitive and instructional design research demonstrated that the crux of algebraic modeling is learning the expression embedding grammar (e.g., putting 800-y and 40x together into 800-40x) [16], and that substitution tasks ("Substitute 500-x for y in y/3") are effective for learning this grammar (KC) [18]. Thus, we introduced composition scaffolding to break down problems according to the underlying cognitive processing (Figure 3a steps 1-3) and explicit practice for the hidden hard KC through isolated steps (Figure 3a step 3) or isolated tasks (Figure 3b). We also introduced multiple

Table 2: Content redesign strategies derived from analytics to create focused tasks.

Analytics about the original tutor	Content redesign strategies for focused tasks
Inappropriate amount of practice on KCs: • Many students over-practiced easier KCs • Many students under-practiced hard KCs • Different KCs needed different # of opp. to mastery	Reduce over-practice on easier KCs and under-practice on hard KCs, e.g., • Eliminate fixed steps of untargeted KCs (i.e., easier KCs or other hard KCs) • Provide dynamic scaffolding
Inadequate scaffolding for hard KCs: • Required too many # of opp. to mastery on hard KCs • No explicit practice on hidden hard KCs	Provide effective scaffolding for hard KCs informed by prior research, e.g., • Composition scaffolding [16] • Explicit practice on hidden hard KCs [18] • Multiple-choice questions for enhancing understanding [26]
Common errors persistent across opportunities and KCs	Provide error feedback and hint messages to address common errors early on

choice for enhancing understanding of hard KCs (Figure 3b), based on prior research on comprehension fostering design [26].

Provide feedback and hint messages to address common errors early on. We found that common errors were persistent across opportunities of a KC and across KCs in the original tutor. We thus introduced error feedback and hint messages in focused tasks especially in focused part tasks, rather than requiring students to do error correction in full tasks where they would experience higher cognitive load. For example, we designed 1-operator focused part tasks (for original 2-operator tasks in Figure 2) with error feedback to address common errors early on in simpler tasks (Figure 4b).

Adaptive tutoring simulation. Building on prior work [13], we created a method that simulates the practice sequence that would be provided by a redesigned tutor (with a refined KC model and redesigned content), and estimates the time required to master a set of KCs, to optimize individualized learning and conduct final refinement of the redesigned tutor. First, we used this simulation to optimize the task selection algorithm by comparing a set of algorithms. We omit details since the idea to convey here is to utilize simulation to help with redesign, rather than to introduce a specific algorithm. Second, we used this simulation to optimize the student model parameters for KCs. We compared three sets of parameters: two were data-tuned with different parameters for different KCs, and the other was hand-set with the same parameters for all KCs (used in the original tutor). Our simulation takes as input a task selection algorithm, a student model with parameters, a KC model, a set of tasks, a strategy to simulate student performance, the number of students, and the assumed seconds per step. It simulates a practice task sequence for each student until the student reaches mastery for all KCs or runs out of tasks. It outputs descriptive statistics of distributions over simulated students of practice minutes, the number of mastered KCs, knowledge levels of KCs, etc. Our simulation suggested the superiority of one of our new selection algorithms and one data-tuned parameter set. With this combination, students' practice time was reduced while they reached a similar or better mastery status, compared to the alternatives.

3 CLASSROOM EXPERIMENTS

3.1 Experimental Design and Setup

We conducted a classroom experiment to investigate whether the redesigned tutor (Data-tuned Adaptive (DA) condition) yields better learning than the original tutor (Control condition). Table 3 lists the comparison between the two conditions. Our experiment aimed at evaluating the overall combined effect of redesigned components, and thus evaluating our multi-method redesign approach. We ran this classroom experiment in the fall of 2019 with high school Algebra 1 classes for one month, with two 40-minute periods on separate days per week. There were eight class periods taught by three teachers across two schools, with two class levels (more advanced or less advanced Algebra I classes). Students were randomly assigned to two conditions within each class period. Students accessed the tutor during their normal class periods, with the teacher providing support as needed. Both conditions followed the same sequence: pretest, practice, posttest. After removing students who were absent in pretest, practice or posttest (missingness was not dependent on the condition), we obtained a sample of 129 students for analysis, with 69 and 60 students in the Control and the DA condition respectively. Regarding demographics, 49% of the students were females, 27% qualified for free or reduced lunch and 11% were Black, Latinx, or multiracial. Both pretest and posttest included two full tasks (e.g., Figure 1) and three unscaffolded focused whole tasks (e.g., Figure 3a without the scaffolding) of the Modeling unit, and four full tasks of the two Explanation units (two from each unit) (e.g., Figure 2). To make sure pretest and posttest were equally difficult, we prepared two forms of tests with different story problems but with the same skill coverage and task order. Students were randomly assigned one form as the pretest and the other as the posttest.

3.2 Analysis Methods

We examined the overall effectiveness of the redesigned tutor by comparing posttest scores (controlling for pretest scores and other factors) between the two conditions. To facilitate understanding the processes leading to the overall effect, we conducted analyses with different focuses to see whether predicted improvements according to our redesign goals were met. We summarize four questions for analyses and explain our analysis methods as follows.

3.2.1 *RQ1: Did the redesigned tutor yield higher learning outcomes?*

To control for factors affecting posttest scores, we compared learning outcomes between the two conditions by multiple regression predicting posttest scores given the pretest scores, condition, pretest form and class level of each student. The direction and significance of the coefficient of the condition indicator shows the effect of our

Table 3: Comparison between two experimental conditions (tutors).

System Component		Control Condition	Data-tuned Adaptive (DA) Condition
KC model		More coarse-grained	More fine-grained
Content	Problem types	Only full tasks	Focused tasks in addition to full tasks
	Problem scaffolding or support	• Static scaffolding in all units • Inductive scaffolding for the Modeling unit [19] • Few feedback and hints for common errors	In focused tasks: • Dynamic, composition scaffolding in all units • More feedback and hints for common errors
	Problem set	Same set of full task stories or situations (focused tasks were derived from them)	
Individualization	Student model	Bayesian Knowledge Tracing (BKT) with skill-specific parameters [12]	
		• Hand-set • Skills share the same set of parameters	• Data-tuned • Different skills have different sets of parameters
	Problem selection	Mastery learning based on BKT knowledge levels and 0.95 mastery threshold per skill (KC)	
		Randomly select a task among unmastered tasks (i.e., tasks with unmastered skills)	• Easier unmastered tasks were assigned higher probabilities to be selected • Students with higher overall proficiency were more likely to skip easier tasks

redesign tutor. We also plotted the mean difference between posttest and pretest scores (i.e., learning gain) with 95% CI per condition, and reported Cohen's *d* for the comparison of these difference scores between two conditions. A posttest or pretest score was obtained by computing the proportion correct of each problem over steps and then computing the average over problems. We included the pretest form and class level because they were significant (or marginally sign.) predictors of posttest scores in all regression models. To see whether the improvements could be due to the increase of practice time, we compared practice time by multiple regression predicting practice time with pretest scores and condition (we included pretest scores because it was a significant predictor in some regression models). We also compared learning outcomes and practice time per unit to see whether there is improvement for each unit, and whether the second round of redesign of the Modeling unit yielded greater improvement than the first round. We also examined the practice time distribution over full and focused tasks.

3.2.2 *RQ2: Did the redesigned tutor reduce over- and under-practice?*

The basis of this analysis is a new KC model that models student learning in both conditions to enable comparison, because each condition had different KC models. We conducted a new round of KC refinement based on the data from this experiment, and selected the KC model with the best AIC value for the data covering both conditions. The set of difficulty factors involved in this new KC model is a superset of those discovered previously. It has a total of 26 skills (KCs). We then estimated over- and under-practice in each condition using this new KC model, following the method used in our redesign process. Here, we fitted a student model to the data of a condition and compared the estimated opportunities to mastery with actual opportunities occurred in the condition. We computed the average number of over- or under-practiced opportunities (over all students and then over all skills) per condition, both as an absolute value and as a percentage of the actual opportunities. We chose AFM [24] as our student model here because on the data from this experiment it consistently outperformed BKT in AIC values, and led to similar estimated opportunities to mastery as BKT. We chose 0.8 as the mastery threshold for AFM since this threshold consistently matched the converted thresholds from commonly used BKT mastery threshold 0.95, using BKT parameters fitted from the same data (i.e., 0.95*(1-*pSlip*)+0.05**pGuess*=0.8).

To examine the predicted improvements that the DA condition should reduce over-practice for easy skills and high level students as well as reduce under-practice for hard skills and low level students, we split skills and students into groups. Under each condition, we split skills into two groups (*Easy* and *Hard*) according to a median split over the fitted AFM skill initial difficulty parameters, and computed the mean of under- or over-practiced opportunities over the skills in each group of a condition; we split students into two groups (*High* and *Low*) according to a median split over students' pretest scores, fitted different AFMs to the subsets of data of each student group, and used each AFM to estimate under- or over-practice for the corresponding student group of a condition.

3.2.3 *RQ3: Did the redesigned tutor lead to a more effective practice experience?*

We utilized learning curves to measure the effectiveness of a practice experience, since they were often used in prior research as a subtle way to measure learning outcomes [25]. We estimated error rates over successive practice opportunities for each skill, and computed the average estimated error rate over skills for each opportunity to obtain an aggregated learning curve. A learning curve with a steeper downward slope (i.e., a steeper decrease of error rate per opportunity) indicates a more effective practice experience. To obtain the estimated error rate of a skill opportunity, we used AFM fitted to the data of each condition. To examine the predicted improvements that the DA condition should lead to a more effective practice experience for hard skills and for both low and high level students, we compared learning curves under each skill group (defined in RQ2) by averaging over skills of a skill group, and compared learning curves under each student group (defined in RQ2) by using AFMs fitted per student group for obtaining estimated error rates. We also examined the initial error rate of a learning curve to see how much students were prepared when encountering a new skill.

3.2.4 *RQ4: Did the redesigned tutor select skills to practice progressing from easier to harder to a greater degree?*

We examined the difficulty progression of each condition as follows: from each student's practice sequence, we obtained all the ordered skill opportunity pairs (where the second skill immediately followed or not), and classified this pair into one of three types: transitioning from an easier to a harder skill (EH), transitioning from a harder to an easier skill (HE), or revisiting the same skill. The relative difficulty between skills was obtained by comparing the estimated probabilities of succeeding on each skill opportunity according to the AFM model fitted to the data of each condition. We computed the frequency of each type and the frequency difference between EH and HE (EH-HE) for each student. We then compared these frequencies especially EH-HE between two conditions. We further ran regression models predicting posttest scores with EH-HE, controlling for the pretest scores, practice time, pretest form and class level (we also added condition as a predictor in the regression on the overall dataset), to see whether higher EH-HE was associated with higher learning outcomes, and to rule out the possibility that this association was caused by pretest score or practice time.

3.3 Results

To confirm that randomization was effective, we found that there were no differences in pretest scores (p=0.9) between conditions. Both conditions produced significant learning gains measured by the difference between posttest and pretest scores (ps<0.001). We now describe results centered on the four research questions introduced above, to provide an overall comparison and understanding of processes leading to the overall results. In all the regression models reported, all variables used the original units and scales.

3.3.1 *RQ1: Did the redesigned tutor yield higher learning outcomes?*

Overall, the redesigned DA tutor led to significantly higher learning outcomes (b=0.05, p=0.046; Cohen's d=0.31). Meanwhile, there was no statistical difference in overall practice time (b=4.10, p=0.40), but the distribution of practice time differed dramatically: the DA condition allocated 78% of practice time on average to focused tasks, replacing much of the full task practice, while the Control condition allocated all time to full tasks (Figure 5). These results demonstrate the overall effectiveness of the redesigned tutor, and suggest the effectiveness of focused task practice.

We then compared learning outcomes and practice time of each unit. As a sanity check, we didn't find any statistical differences in pretest scores (ps>0.43) or ceiling effects in pretest or posttest scores per unit. Compared to the Control condition, on the Modeling unit, the most difficult unit (according to posttest scores), we found that the DA condition yielded significantly higher learning outcomes (b=0.08, p=0.013) with no statistical difference in practice time (b=4.15, p=0.39); the DA condition allocated significantly more time to the 2-operator Explanation unit (b=12, p=0.003), and led to higher learning outcomes albeit lacking statistical significance (b=0.06, p=0.17); on the 3-operator Explanation unit, students in the DA condition spent significantly less time (b=-12, p<0.001; 46% less practice time on average) with no statistical difference in learning outcomes (b=-0.004, p=0.92). Looking into practice time distribution, the DA condition allocated the major portion of time to focused tasks while the Control condition allocated all time to full tasks across the three units (Figure 6). These results show that the redesigned tutor led to improvement in each unit either in learning outcomes or efficiency. In particular, the second round of redesign of the Modeling unit yielded greater improvement over the Control condition compared to the first round [17]. The redesigned tutor led to improvement on learning outcomes or efficiency to a lesser degree in the 2-operator Explanation unit, but yielded significantly higher learning efficiency on the 3-operator Explanation unit with almost no full task practice and significantly less practice time in this unit compared to the Control condition. This suggests that the value of focused task practice in the 2-operator Explanation unit was in promoting better knowledge transfer for future learning.

3.3.2 *RQ2: Did the redesigned tutor reduce over- and under-practice?*

As shown in Figure 7 left panel, compared to the Control condition, the DA condition reduced the amount of over-practice by half on average per student per skill, for easy skills and for high pretest students. Specifically, the DA condition dropped the degree of over-practice from 41% (4.59/11.08) to 31% (1.99/6.50) of actual average opportunities for easy skills, and from 28% (3.40/12.30) to 18% (1.55/8.43) for high pretest students. Although the numbers of over-practiced opportunities were not statistically significantly different (easy skills: t(24)=1.3, p=0.21; high pretest students: t(50)=1.4, p=0.16), we found a medium effect size for easy skills (Cohen's d=0.51; high pretest students: d=0.40), and practical significance of the reduction for both groups: considering the sum of reduced over-practiced opportunities over skills for easy skills ((4.59-1.99)*13=34) and for high pretest students ((3.40-1.55)*26=48), the overall reduction in over-practice was roughly the equivalence of opportunities for *at least one more mastered skill* in the span of the study.

As shown in Figure 7 right panel, there was a significant amount of under-practice in both conditions. We found that the average total practice time students had was only around 64% of the planned total practice time, which may be the main reason. Still, the DA condition led to a reduction of around a quarter of under-practiced opportunities on average per student per skill for hard skills and for low pretest students, compared to the Control condition. The numbers of under-practiced opportunities were statistically significantly different between the conditions with medium or large effect sizes (hard skills: t(24)=2.2, p=0.04, d=0.86; low pretest students: t(50)=2.2, p=0.03, d=0.62). In terms of practice significance of this reduction, the sum of reduced under-practiced opportunities over skills for hard skills ((26.4-19.2)*13=94) and for low pretest students ((23.9-18)*26=153) were non-trivial.

Altogether, the results show that the DA condition reduced over-practice for easy skills and high level students, and reduced under-practice for hard skills and low level students, compared to the Control condition, meeting our predicted improvements.

3.3.3 *RQ3: Did the redesigned tutor lead to a more effective practice experience?*

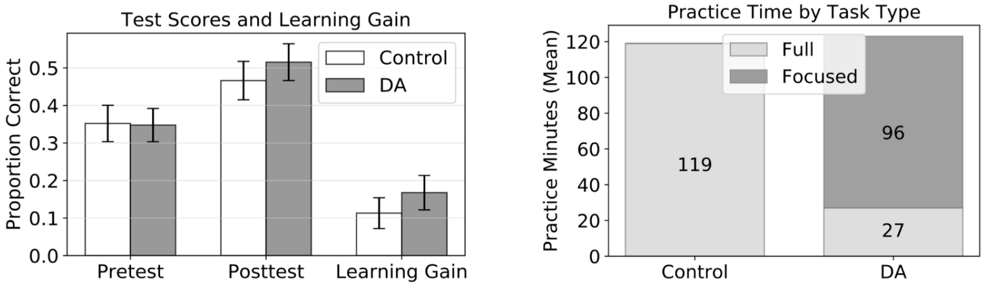


Figure 5: The Data-tuned Adaptive condition produced higher learning outcomes than the Control condition through replacing much of the full task practice in the original tutor with focused task practice. Error bars represent 95% CIs.

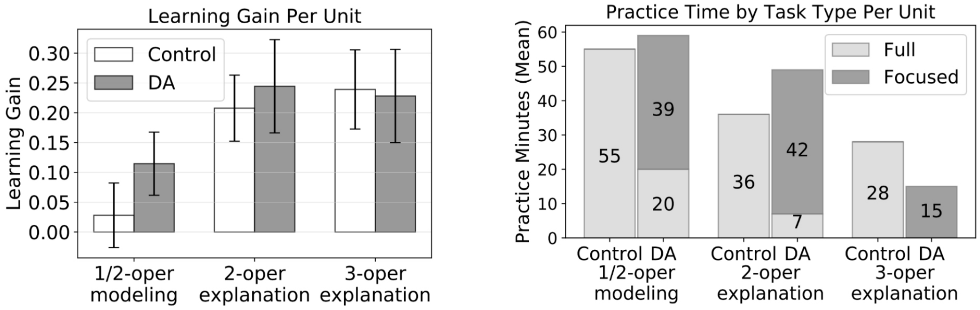


Figure 6: The Data-tuned Adaptive condition led to higher learning outcomes in 1/2-operator Modeling and 2-operator Explanation units, and equivalent learning outcomes in 3-operator Explanation unit (left panel) despite needing much less time in 3-operator Explanation unit (right panel), compared to the Control condition. Error bars represent 95% CIs.

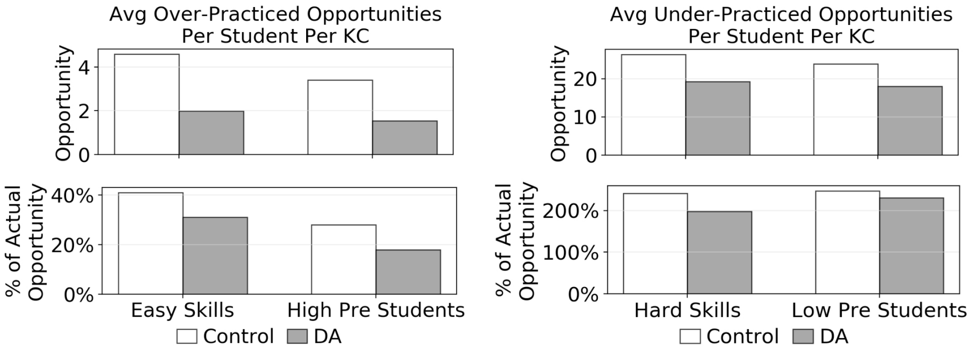


Figure 7: The Data-tuned Adaptive condition reduced over-practice for easy skills and high pretest level students, and reduced under-practice for hard skills and low pretest level students, compared to the Control condition.

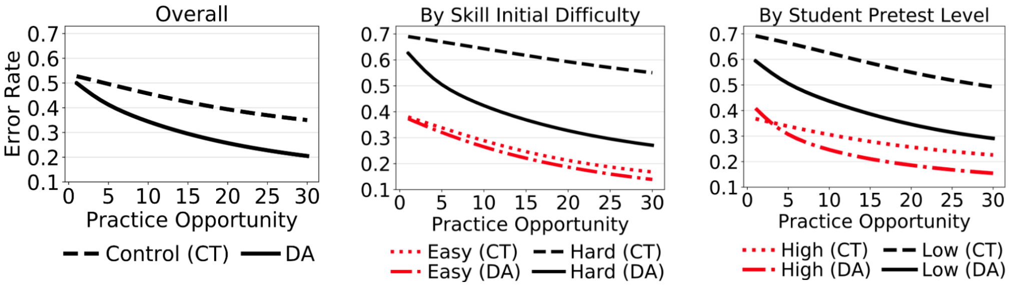


Figure 8: The Data-tuned Adaptive condition yielded more effective practice experiences (i.e., steeper downward slopes) for hard skills and both levels of students, compared to the Control condition.

Overall, the DA condition led to a much steeper downward learning curve as shown in Figure 8. Examining different skill groups, the DA condition led to much faster decrease in error rates for hard skills and a similar speed of decrease for easy skills (Figure 8 middle panel); examining different student groups, the DA condition led to faster decrease in error rates for both levels of students and the decreasing speed for low level students was pronounced (Figure 8 right panel). These results suggest that the DA condition provided more effective practice experiences for hard skills and for both levels of students, meeting our predicted improvements. Moreover, the initial error rates for hard skills and for low level students in the DA condition were around 0.1 lower than those of the Control condition, suggesting that students were generally more prepared when encountering a new hard skill, and low level students were more prepared when encountering a new skill.

3.3.4 *RQ4: Did the redesigned tutor select skills to practice progressing from easier to harder to a greater degree?*

As shown in Figure 9, the DA condition led to a higher frequency of easier-to-harder skill progression (EH), and a lower frequency of harder-to-easier skill progression (HE), doubling the frequency difference (EH-HE) from 10% more likely to 20% more likely to choose EH over HE, and this difference in progression was practically and statistical significant ($t(127)$=3.2, p=0.002; Cohen's d=0.57). One might argue that the HE progression indicates a tutor's intention to lower the difficulty which might be beneficial in some cases, yet such an adjustment indicates that the tutor has selected a skill too difficult in the first place, or switches to another skill before the current skill becomes easy (i.e., mastered) for a student. The HE progression is harmful or suboptimal in that it is at the cost of opportunities where students could learn better from other types of progression (EH or Same). To confirm the benefit of EH progression over HE progression, the frequency difference (EH-HE) was a significant predictor for posttest scores when controlling for pretest scores and practice time (and other factors) for each condition and overall (Table 4). This also suggests the separate contribution of this task selection feature on learning outcomes, although we admit that students' proficiency (which might not be fully represented by pretest scores) could still be an alternative explanation for this association. These results show that the redesigned tutor selected skills to practice progressing from easier to harder to a greater degree and this feature was associated with better learning outcomes.

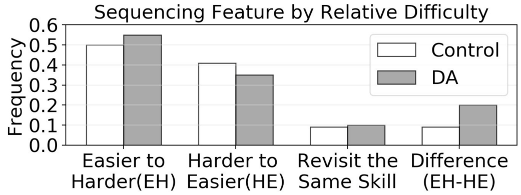


Figure 9: The Data-tuned Adaptive condition selected skills progressing from easier to harder to a greater degree (i.e., higher EH-HE), compared to the Control condition.

Table 4: Overall and per condition regression show that higher frequency differences between easier-to-harder and harder-to-easier skill progression (EH-HE) was associated with higher posttest scores (controlling for other factors).

All	Control	DA
b=0.30, p<0.001	b=0.29, p=0.005	b=0.35, p=0.03

4 DISCUSSION AND CONCLUSION

In this paper, we demonstrate a multi-method approach (called MADDRED) to data-driven redesign of tutoring systems, and provide empirical evidence of its effectiveness and generality through a classroom experiment, extending our prior approach and evaluation [17]. The key feature of this approach is to identify hard skills and provide effective and efficient practice on them through focused tasks and optimized task selection. Our classroom experiment shows that compared to the original tutor, the redesigned tutor led to significantly higher learning outcomes, reduced over- and under-practice, yielded a more effective practice experience, and selected skills progressing from easier to harder to a greater degree. Regarding practice time distribution, the redesigned tutor replaced much of the full task practice in the original tutor with focused task practice. Thus these results also provide indirect evidence for the effectiveness of our focused tasks.

Our approach systematically combines new and existing learning analytics and instructional design methods. We created new analytics methods that increase efficiency of prior methods, such as Difficulty Factor Effect Analysis for KC model refinement, and Probability-Propagated Practice Estimation for estimating opportunities to mastery, amount of over-/under-practice. We also created a data-driven instructional design method, Focused Practice Task Design, with content redesign strategies derived from analytics of opportunities to mastery and errors, informed by prior research.

Although the redesigned tutor reduced under- and over-practice, it still led to some amount of over-practice and a non-trivial amount of under-practice. Although there might be benefits from a small amount of over-practice, this time might still be better spent on other under-practiced skills. Thus, we may need redesign efforts to further reduce over-practicing. For example, refining the KC model and student model parameters based on the newly collected data may lead to more accurate knowledge estimates (and design), and help reduce over-practicing. As for under-practice in the redesigned tutor, the primary reason may be the discrepancy between the actual practice time and the planned time for a variety of reasons including students' starting late, signing off early. Although we might consider a longer span study or a smaller set of target skills, we may consider making the scaffolding design more adaptive to students' differences. After all, data-driven redesign is intended as an iterative process. In addition, examining students' learning outcomes in a delayed test or standardized tests may provide more evidence regarding the effect of the redesign on robust learning.

As an initial test of the generality of the approach, we tested it in multiple units in the algebra task domain in the same intelligent tutoring system. To further test the generality, we need to apply it to other task domains or other tutoring systems. Our approach is intended to be applicable to systems with learning-by-doing activities, and where activities and instructions are designed and organized based on a KC model. In systems with other types of activities (e.g., reading, video watching), analytics on engagement (e.g., time or usage on resources [30]) might provide valuable insights into redesign. Moreover, our data-driven redesign approach can borrow from human-centered learning analytics to answer design questions that data analytics alone cannot answer, and it may also enhance human-centered learning analytics processes.

Our study investigated how to combine methods to maximally improve a tutor, and whether our systematic combination leads to student learning improvements. We believe that such an approach and evaluation are of value for instructional design practices and learning engineering research. Our experiment does not allow teasing apart the effects of different components; there may be value in further studies to isolate the contributions of these components.

The present research serves as a first step towards understanding how to combine methods for data-driven redesign of tutoring systems and courses. Our work provides general guidance on how to convert learning analytics into better system design, an important need in LAK research and practice. Our work may also help define and enhance data-driven learning engineering processes.

ACKNOWLEDGMENTS

This work was supported by Bill and Melinda Gates Foundation Prime Award #OPP1196889. We also thank all reviewers for their valuable feedback which helped us improve our manuscript.

REFERENCES

[1] Vincent Aleven and Kenneth R Koedinger. 2013. Knowledge component (KC) approaches to learner modeling. *Design Recommendations for Intelligent Tutoring Systems* 1 (2013), 165–182.

[2] Vincent Aleven, Elizabeth A McLaughlin, R Amos Glenn, and Kenneth R Koedinger. 2016. Instruction based on adaptive learning technologies. *Handbook of research on learning and instruction* (2016), 522–560.

[3] Vincent Aleven and Jonathan Sewall. 2016. The frequency of tutor behaviors: a case study. In *International Conference on Intelligent Tutoring Systems*. Springer, 396–401.

[4] John R Anderson, Albert T Corbett, Kenneth R Koedinger, and Ray Pelletier. 1995. Cognitive tutors: Lessons learned. *The journal of the learning sciences* 4, 2 (1995), 167–207.

[5] Kimberly E Arnold and Matthew D Pistilli. 2012. Course signals at Purdue: Using learning analytics to increase student success. In *Proceedings of the 2nd international conference on learning analytics and knowledge*. 267–270.

[6] Aneesha Bakharia, Linda Corrin, Paula De Barba, Gregor Kennedy, Dragan Gašević, Raoul Mulder, David Williams, Shane Dawson, and Lori Lockyer. 2016. A conceptual framework linking learning design with learning analytics. In *Proceedings of the sixth international conference on learning analytics & knowledge*. 329–338.

[7] Hao Cen, Kenneth Koedinger, and Brian Junker. 2006. Learning factors analysis–a general method for cognitive model evaluation and improvement. In *International Conference on Intelligent Tutoring Systems*. Springer, 164–175.

[8] Hao Cen, Kenneth R Koedinger, and Brian Junker. 2007. Is Over Practice Necessary?-Improving Learning Efficiency with the Cognitive Tutor through Educational Data Mining. *Frontiers in artificial intelligence and applications* 158 (2007), 511.

[9] Zhongzhou Chen, Christopher Chudzicki, Daniel Palumbo, Giora Alexandron, Youn-Jeng Choi, Qian Zhou, and David E Pritchard. 2016. Researching for better instructional methods using AB experiments in MOOCs: results and challenges. *Research and Practice in Technology Enhanced Learning* 11, 1 (2016), 1–20.

[10] Benoît Choffin, Fabrice Popineau, Yolaine Bourda, and Jill-Jênn Vie. 2019. DAS3H: Modeling Student Learning and Forgetting for Optimally Scheduling Distributed Practice of Skills. (2019).

[11] Doug Clow. 2012. The learning analytics cycle: closing the loop effectively. In *Proceedings of the 2nd international conference on learning analytics and knowledge*. 134–138.

[12] Albert T Corbett and John R Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction* 4, 4 (1994), 253–278.

[13] Shayan Doroudi, Vincent Aleven, and Emma Brunskill. 2017. Robust evaluation matrix: Towards a more principled offline exploration of instructional policies. In *Proceedings of the fourth (2017) ACM conference on learning@ scale*. 3–12.

[14] K Anders Ericsson et al. 2006. The influence of experience and deliberate practice on the development of superior expert performance. *The Cambridge handbook of expertise and expert performance* 38, 685-705 (2006), 2–2.

[15] José González-Brenes, Yun Huang, and Peter Brusilovsky. 2014. General features in knowledge tracing to model multiple subskills, temporal item response theory, and expert knowledge. In *The 7th international conference on educational data mining*. 84–91.

[16] Neil T Heffernan and Kenneth R Koedinger. 1997. The composition effect in symbolizing: The role of symbol production vs. text comprehension. In *Proceedings of the nineteenth annual conference of the cognitive science society*. 307–312.

[17] Yun Huang, Vincent Aleven, Elizabeth McLaughlin, and Kenneth Koedinger. 2020. A General Multi-method Approach to Design-Loop Adaptivity in Intelligent Tutoring Systems. In *International Conference on Artificial Intelligence in Education*. Springer, 124–129.

[18] Ken Koedinger and Elizabeth McLaughlin. 2010. Seeing language learning inside the math: Cognitive analysis yields transfer. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 32.

[19] Kenneth R Koedinger and John R Anderson. 1998. Illustrating principled design: The early evolution of a cognitive tutor for algebra symbolization. *Interactive Learning Environments* 5, 1 (1998), 161–179.

[20] Kenneth R Koedinger, Ryan SJd Baker, Kyle Cunningham, Alida Skogsholm, Brett Leber, and John Stamper. 2010. A data repository for the EDM community: The PSLC DataShop. *Handbook of educational data mining* 43 (2010), 43–56.

[21] Kenneth R Koedinger, Jihee Kim, Julianna Zhuxin Jia, Elizabeth A McLaughlin, and Norman L Bier. 2015. Learning is not a spectator sport: Doing is better than watching for learning from a MOOC. In *Proceedings of the second (2015) ACM conference on learning@ scale*. 111–120.

[22] Jung In Lee and Emma Brunskill. 2012. The Impact on Individualizing Student Models on Necessary Practice Opportunities. *International Educational Data Mining Society* (2012).

[23] Ran Liu and Kenneth R Koedinger. 2017. Closing the Loop: Automated Data-Driven Cognitive Model Discoveries Lead to Improved Instruction and Learning Gains. *Journal of Educational Data Mining* 9, 1 (2017), 25–41.

[24] Marsha Lovett, Oded Meyer, and Candace Thille. 2008. The Open Learning Initiative: Measuring the Effectiveness of the OLI Statistics Course in Accelerating Student Learning. *Journal of Interactive Media in Education* (2008).

[25] Brent Martin, Antonija Mitrovic, Kenneth R Koedinger, and Santosh Mathan. 2011. Evaluating and improving adaptive educational systems with learning curves. *User Modeling and User-Adapted Interaction* 21, 3 (2011), 249–283.

[26] Santosh A Mathan and Kenneth R Koedinger. 2002. An empirical assessment of comprehension fostering features in an intelligent tutoring system. In *International Conference on Intelligent Tutoring Systems*. Springer, 330–343.

[27] Roxana Moreno and Richard E. Mayer. 2010. *Techniques That Increase Generative Processing in Multimedia Learning: Open Questions for Cognitive Load Research.* Cambridge University Press, 153?178. https://doi.org/10.1017/CBO9780511844744.010

[28] Behrooz Mostafavi and Tiffany Barnes. 2017. Evolution of an intelligent deductive logic tutor using data-driven elements. *International Journal of Artificial Intelligence in Education* 27, 1 (2017), 5–36.

[29] Mitchell J Nathan, Kenneth R Koedinger, Martha W Alibali, et al. 2001. Expert blind spot: When content knowledge eclipses pedagogical content knowledge. In *Proceedings of the third international conference on cognitive science*, Vol. 644648.

[30] Quan Nguyen, Michal Huptych, and Bart Rienties. 2018. Linking students' timing of engagement to learning design and academic performance. In *Proceedings of the 8th international conference on learning analytics and knowledge*. 141–150.

[31] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. In *Advances in neural information processing systems*. 505–513.

[32] John C Stamper and Kenneth R Koedinger. 2011. Human-machine student model discovery and improvement using DataShop. In *International Conference on Artificial Intelligence in Education*. Springer, 353–360.

[33] Guojing Zhou, Jianxun Wang, Collin F Lynch, and Min Chi. 2017. Towards Closing the Loop: Bridging Machine-Induced Pedagogical Policies to Learning Theories. *International Educational Data Mining Society* (2017).