Machine Analysis of Students' Mathematical Representations for Multiplication and Division Problems

by

Matthew Peairs
S.B. Electrical Engineering and Computer Science
Massachusetts Institute of Technology, 2009

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of Master of Engineering in Electrical Engineering and Computer Science at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2013

Author

....

Department of Electrical Engineering and Computer Science
August 30, 2013

Certified by

....

Dr. Kimberle Koile
Research Scientist, MIT CECI
Thesis Supervisor
August 30, 2013

Accepted by

....

Prof. Albert R. Meyer
Chairman, Master of Engineering Thesis Committee

Machine Analysis and Sorting of Students' Mathematical Representations for Multiplication and Division Problems

By
Matthew Peairs

Submitted to the Department of Electrical Engineering and Computer Science on August 30, 2013 in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

This project extends Classroom Learning Partner classroom interaction software to include a semantic interpretation component. This semantic interpretation, combined with existing syntactic interpretation, enables the software to tag and group student work using knowledge of the math used in both creating and solving problems. The analysis is being prototyped using student work in grades 4 and 5, with focus on multiplication and division. First, during the authoring step, the notebook author gives each page a "page definition" that encapsulates the mathematical problem presented on that page. For a multiplication or division problem, this involves setting the three numbers connected by the product relation (e.g., 6 * 3 = 18), marking which of those numbers are given by the problem or otherwise unknown, and selecting an overall context for the problem, such as equal groups or area. Then, once students have submitted their work, the analysis component takes the raw output of the syntactic interpretation step and relates it back to the mathematical content of the page to assign each student's work a set of automatically generated tags. These tags address the correctness of a student's methods and results, as well as highlighting different problem-solving strategies that students might have used to arrive at the same answer. Finally, the teacher can sort student submissions by these various tags to quickly find noteworthy or contrasting examples to present to the class.

Thesis Supervisor: Dr. Kimberle Koile
Title: Research Scientist, MIT CECI

Contents

1 Introduction

Classroom Learning Partner (CLP) is a tablet-PC-based classroom interaction system designed and built to investigate pen-based interaction and wireless communication in K-12 education [2, 4]. One of its goals over the past several years has been to design, implement, and test in classrooms a suite of software tools that enable students to use the tablet's pen to create mathematical representations and wirelessly share their representations with their teacher. This immediate feedback allows the teacher to structure the class discussion around student work. He or she can select and share with the entire class examples that feature different representations or problem-solving strategies. The teacher also can identify students who may be struggling or pinpoint concepts that the class as a whole needs to spend additional time reviewing. But how does the teacher sift through an entire class's work to identify pedagogically interesting examples or to see the underlying patterns in students' understanding of the material? As we have observed in classrooms, it can be difficult for a teacher to make these judgments in real time, all while continuing to teach and engage the students. But what if the students' work could be automatically analyzed and sorted into categories based on students' mathematical representations and methods of reasoning?

This thesis project is the design and implementation of such an analysis and sorting system, and as such has two primary goals: to provide a flexible and extensible way to represent the wide variety of problems that might be part of an upper elementary math curriculum, for multiplication and division in particular; and to develop methods to automatically analyze student work and present the teacher with the most interesting and salient aspects of different students' submissions. To accomplish these goals, this project ties together CLP's components and adds a critical new component. CLP's existing components support students' creation and wireless sharing of mathematical representations, syntactic analysis of the representations, and a teacher's viewing of the representations. The critical new component is that of semantic analysis and sorting of the representations based on the mathematical content of the students' work. Prior to this project, CLP, for example, could analyze a student's mathematical representation and produce a syntactic description such as “four groups of three,” but it could not show the teacher that description nor produce a semantic description stated in terms of the mathematics, e.g., whether the “four groups of three” was a correct mathematical representation for the problem or whether the student was nearly correct but represented “three groups of four” instead.

A teacher's time is a valuable and limited resource, especially in a large classroom setting. It is our hope that, by analyzing each student's individual work and making the results of that analysis easy to access and understand, this project will allow teachers to more efficiently assess their students' progress and more effectively structure lessons and discussions.

1.1 Examples

In recent classroom trials, elementary-school teachers have been using CLP to teach their students about multiplication and division in various forms and contexts. For example, a particular problem may invite the student to show his or her work using one of a number of different representations, such as arrays or “number sentences” (equations). Some problems may simply entail multiplying two known factors to find an unknown product, while others may have numerous possible solutions subject to certain constraints—such as asking students to depict a group of birds and spiders that have a total of 30 legs. A story problem might use division in the context of objects divided into equal groups, or distance traveled over time, or a simple scale factor ("Alice is half as old as Bob..."). The challenge, then, is to build a framework that supports all of these different problems and that can recognize the important differences between them. Shown in Figures 1-1 through 1-6 below are examples of 4th grade student work that, as a result of this thesis project, CLP can now semantically interpret, sort by grouping into similarity classes, and present to a teacher for viewing.

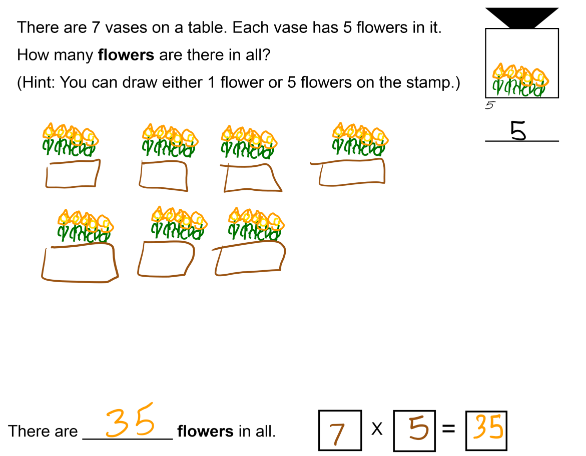

Figure 1-1. Multiplication problem. In the upper-right corner of the page is what we call a stamp; this tool allows students to place multiple copies of an image on the page. See Section 2.1 for further details.)

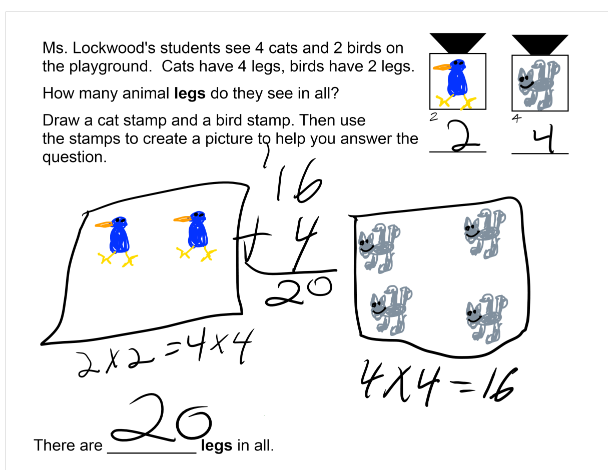

Figure 1-2. Multiplication problem involving adding results of sub-problems

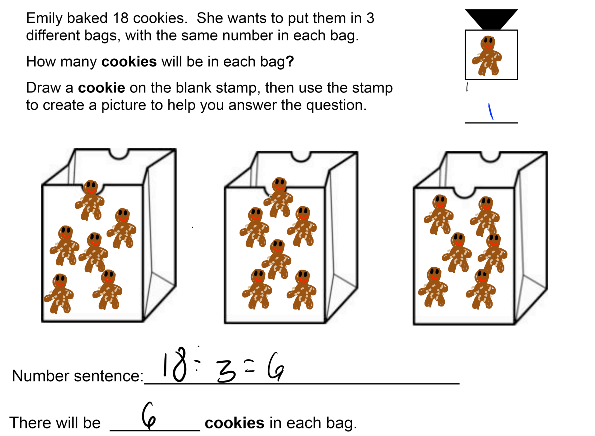

Figure 1-3. Division problem: "dealing out" to figure out the size of a group

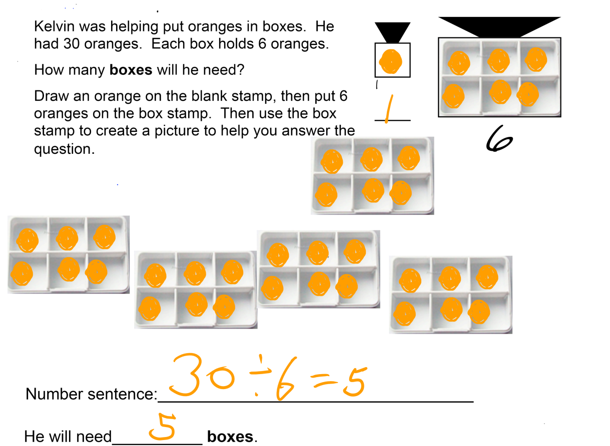

Figure 1-4. Division problem: grouping using a "container" to figure out the number of groups

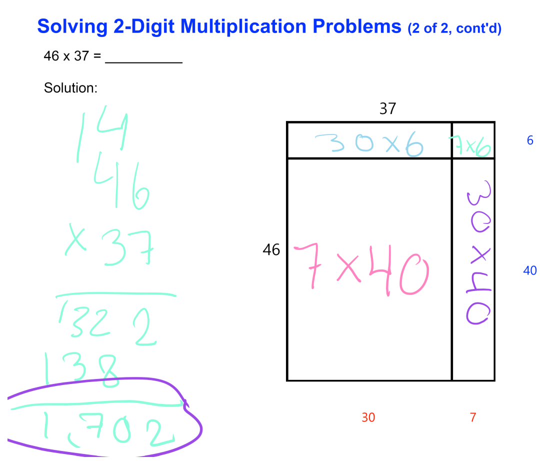

Figure 1-5. Using an array for multi-digit multiplication

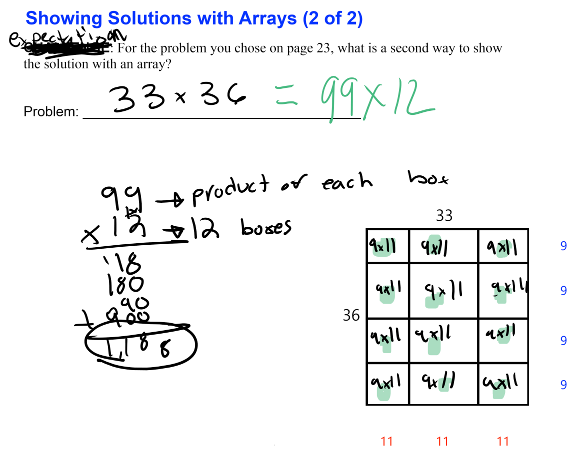

Figure 1-6. Another multiplication problem with arrays, showing a different strategy

1.2 Outline

Chapter 2 describes CLP, which provides the infrastructure for this thesis project. Chapter 3 describes the design goals of this project. Chapter 4 provides the details of the project's implementation. Chapter 5 discusses the results of the project and provides examples of the analysis in action. Chapter 6 describes possible ways to build upon this work in the future. Chapter 7 contains a list of references.

2 Background

2.1 Technology: CLP Software

The analysis and sorting tool described in this thesis is an extension of our existing classroom interaction system called Classroom Learning Partner (CLP) [2, 4]. During a classroom session, the teacher and each student have a tablet computer, and a tablet computer is connected to a projector, creating a public display with which student and teacher machines communicate. Lessons are organized in electronic notebooks, which teacher and students open via CLP. Each student writes in his or her notebook by selecting a page in the notebook and using the tablet computer's pen to write on the tablet's screen. Students have a variety of colors and pen widths available, as well as math tools such as arrays. After writing, students wirelessly submit their "digital ink" answers to the teacher. (See Figure 2-1 for a screen shot the student UI.) The teacher can view all student submissions and choose a subset to display anonymously for the class, i.e., without student names visible, via the projector. He or she can display multiple examples of student work simultaneously and annotate the examples, using the tablet pen, as a way to encourage conversations in which students defend their reasoning and listen critically to others' reasoning. (See Figure 2-2 for a screen shot of the teacher UI.)

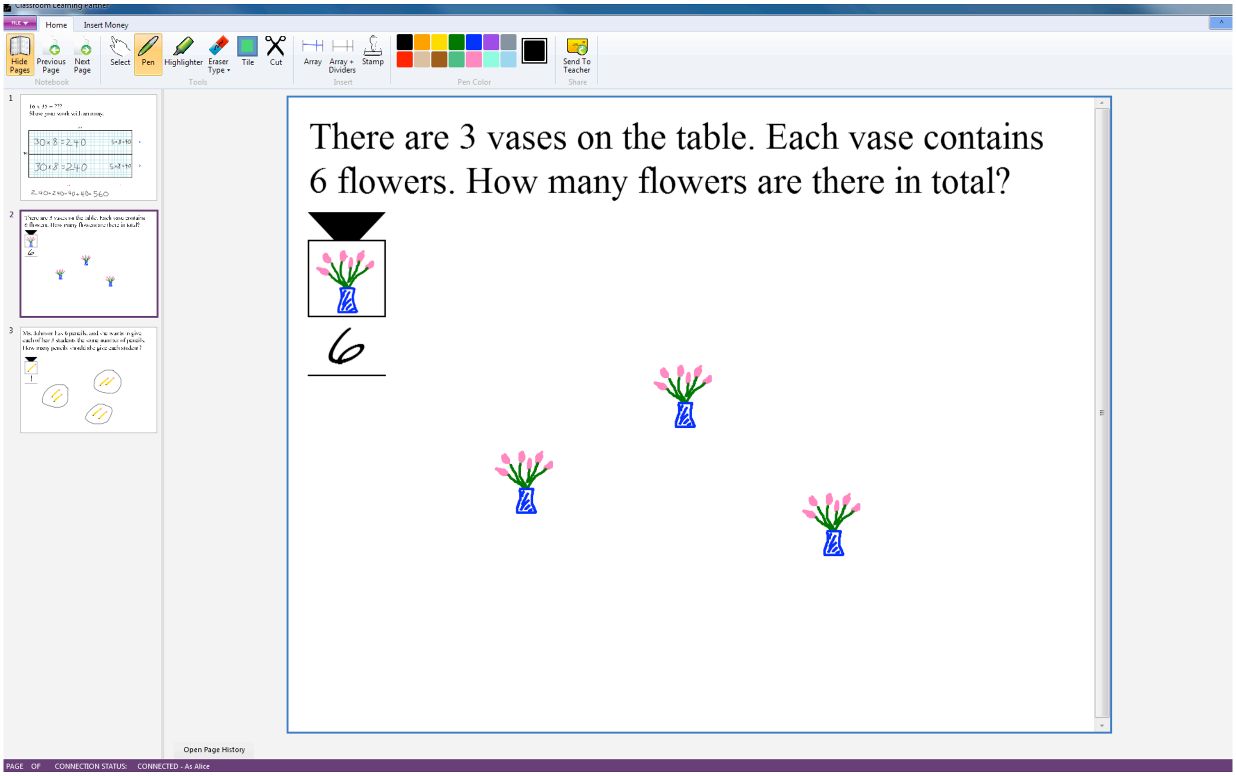

Figure 2-1: Student tablet UI; thumbnails of notebook pages are shown in the left panel, students work in the main display window

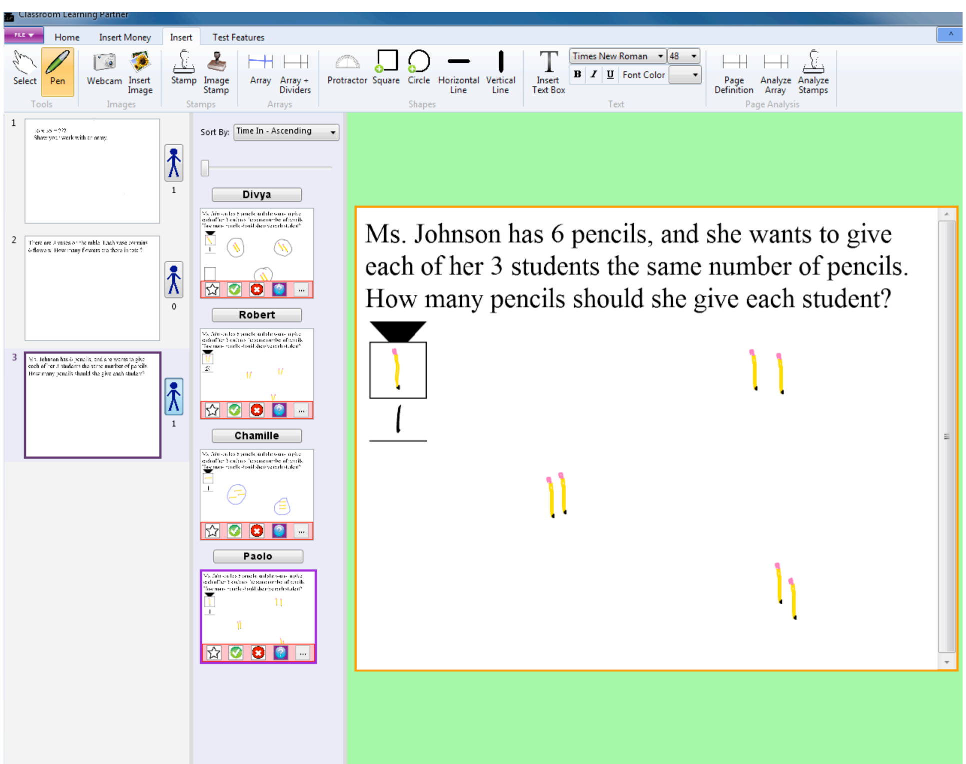

Figure 2-2: Teacher tablet user interface; student submissions for a particular page are displayed in a panel between the notebook page display and the main window

This project, as the existing CLP infrastructure upon which it builds, is written in C#, using the Catel framework for ease of serialization and binding data structures to UI controls that are built using the Windows Presentation Foundation (WPF).

CLP is based on the idea of a “notebook,” which corresponds to a particular lesson. Each notebook consists of a number of “pages,” each of which typically features an exercise for the student to work. Everything the student adds to the page—using tools such as ink, arrays, or stamps—is represented as a “page object.” As the student progresses through the notebook, he or she can submit individual pages to the teacher's machine.

The overall lifecycle of a CLP notebook page involves several steps:

1. Prior to class, the page author (who may be a teacher or a curriculum designer) creates a new notebook page, adding text, images, and other features.
2. During class, the student opens the page, works through the problem, and submits their results to the teacher.
3. The student's work is tagged with information such as student name and time of submission and automatically interpreted syntactically, though prior to this thesis project the information was stored in a log file for researcher use and not stored with the page.
4. When the teacher has received submissions from multiple students for a particular page, he or she can sort and group the students' pages using any of the available tags.

This thesis involves extensions to steps 1, 3, and 4: In step 1, the author uses a new authoring feature introduced in this project to assign the page a “page definition” that encapsulates the mathematical problem being presented on the page. (See Section 4.1 for details.) In step 3, the student work is now interpreted semantically as well as syntactically, and both interpretations are stored on a page using what we call tags. The new semantic tags are used to indicate interesting features of that solution, such as correctness or the use of a particular problem-solving strategy. Finally in step 4, the teacher's sorting and grouping UI now includes tags associated with interpretation of the student work.

There are several tools that allow the student to add page objects to a page. By selecting the pen tool, the student can freely draw images using the tablet's stylus. Each individual stroke is attached to the page as a page object. Ink strokes are not used on their own for the purposes of automated page analysis, although the stamp-grouping code makes use of ink page objects, as noted below.

The stamp tool allows a student to create a custom-drawn image and place multiple copies of that image on the page. It enables students to draw freehand, but with enough structure for machine analysis of the resulting representation [2, 3]. Each stamp copy is attached to the page as a page object. Before placing copies of a stamp, the student must input the number of “parts” represented by that stamp. This information allows the analysis methods to determine whether each stamp is meant to represent one element of a group, or an entire group at once. For some problems, the notebook author may provide what we call a “container stamp,” which provides an additional way to visually represent groups of items—for instance, a container stamp may represent a vase that contains several flowers, or a crate that contains some number of oranges.

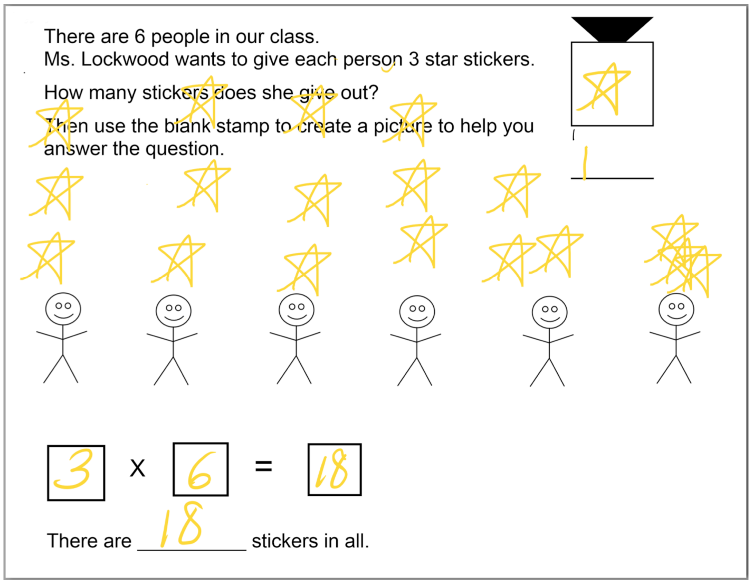

Figure 2-3: A stamp that represents a single part (in this case, a star).

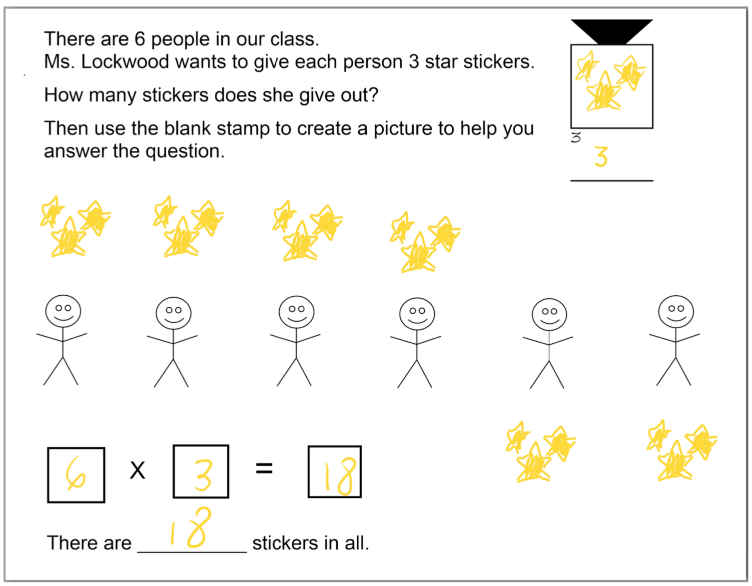

Figure 2-4: A stamp that represents multiple parts.

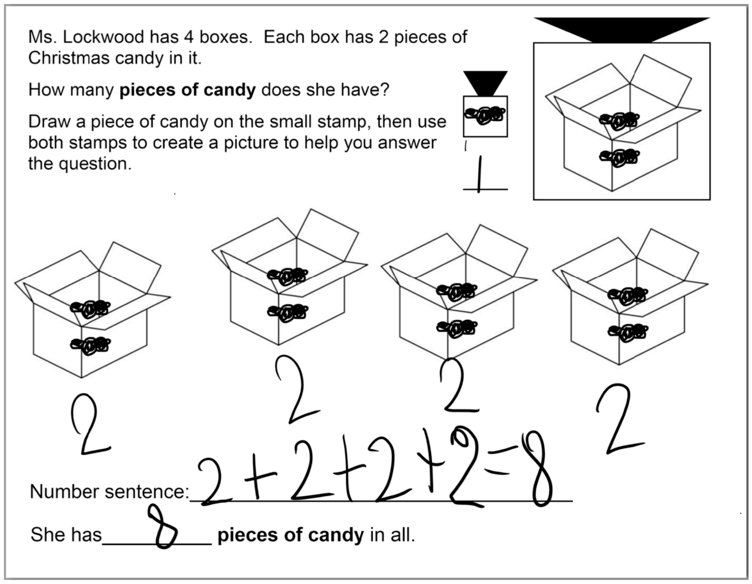

Figure 2-5: An example of a container stamp in action. The notebook author created the container stamp (the box) while writing the page. In the classroom, the student drew a piece of candy on the smaller stamp and stamped it twice onto the container stamp, then used the container stamp to depict four boxes with two candies each.

Once the student has placed stamps on the page and submitted his or her work to the teacher, CLP can analyze the arrangement of stamps to infer how the student intended them to be grouped [3]. The software takes into account both the location of stamps on the screen (nearby stamps are likely to be part of the same conceptual group) and ink strokes placed by the student to separate stamps into groups. Container stamps, if they are used on the page, are also taken into account. The work in this thesis builds on the results of that interpretation, relating its output back to the mathematical domain by comparing it to the multiplication or division problem originally defined by the page.

The array tool provides an alternative representation for multiplication and division problems. When creating an array, the student specifies the number of rows and columns and a rectangle with those dimensions appears on the page, represented internally as yet another type of page object. The student then can split the array along its horizontal or vertical axes, or else "snap" together multiple arrays. After splitting an array by using internal dividers, the student can enter numbers to show how they are breaking the array into smaller pieces (see figure 2-6 below).

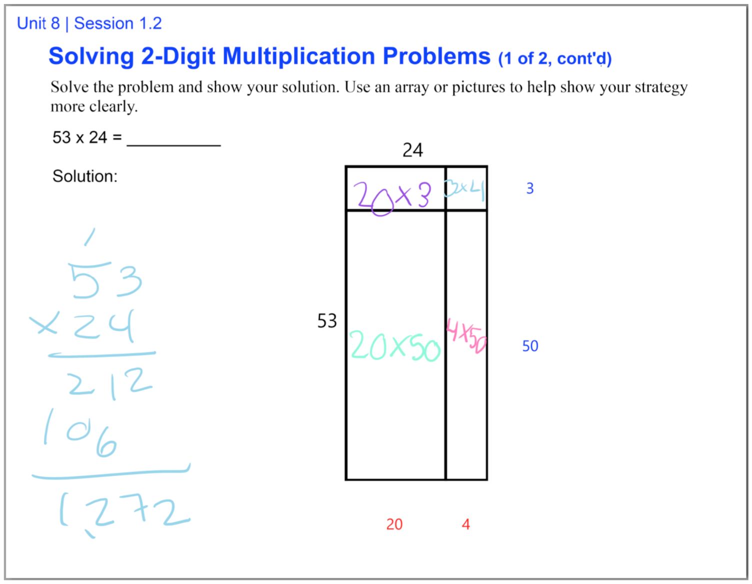

Figure 2-6: The array tool. This student added one horizontal and one vertical divider to split a 24x53 array into four smaller regions.

2.2 Education: Teaching Multiplication and Division

Multiplication and division are inverse operations; a mathematical relationship expressed as multiplication can be reframed as division, and vice versa. For example, consider the following multiplication problem: “There are 3 houses on this street, and each house has 5 windows. How many windows are there in all?” This problem can be represented with a multiplication statement, 3 x 5 = 15. A related division problem is “Each house on this street has 5 windows, and there are 15 windows total. How many houses are there?”, and can be represented using the division operator as $15 \div 5 = 3$. Both problems are asking questions about the same underlying model: 3 houses with 5 windows apiece. In general, simple problems such as these, whether they are coded as multiplication or division, are characterized by *product relations* consisting of two factors and their product. In both of the above examples, the factors are 3 and 5 and the product is 15. The key difference between a “multiplication problem” and a “division problem” is which of the product relation's three values is left for the solver to determine. If the two factors are given, the product is found by multiplying them. If one factor and the product are given, the other factor is found by dividing the product by the given factor. Because multiplication and division can be used to describe the same types of situations, it is beneficial to students' comprehension of the material if the two operations are introduced together, with an emphasis on the relationship between them.

When considering how students might use the CLP tools outlined in Section 2.1 to create representations of a product relation, it's useful to consider the different ways to model multiplication and division in familiar, real-world situations. As outlined in the Common Core State Standards [1], the following is a categorization of different kinds of multiplication and division problems:

1. **Fair Sharing**. “Each house on this street has 5 windows, and there are 15 windows total. How many houses are there?”
2. **Equal Groups**. “There are 25 students into the class, and the teacher divides them into 5 teams of equal size. How many students are on each team?”
3. **Arrays**. “The movie theatre has 90 seats arranged into rows of 10 seats each. How many rows of seats are there?”
4. **Area**. “How many square feet are in a 10-foot by 15-foot room?”
5. **Rate**. “If I drive at thirty miles per hour for two hours, how far will I travel?”
6. **Cost**. “If I have ten dollars and an apple costs two dollars, how many apples can I buy?”
7. **Unit Conversion**. “If there are twelve inches in a foot, how many inches are in six feet?”

8. **Scaling**. “If Alice is 4 years old and Bob is twice as old as Alice, how old is Bob?”
9. **Cartesian Product**. “If Alice has 3 shirts and 4 pairs of pants, how many different outfits can she make?”

In the next section, we describe the extensions to CLP that would support students in solving problem types 1-4. Problem types 5-9 are outside the scope of this thesis but are discussed in Section 5.

3 Design Goals

The ultimate goal of this thesis project is to extend CLP to support the machine analysis of multiplication and division problems. Many of the types of problems listed in the previous section can be addressed using the tools currently available in CLP. The stamp tool lends itself most naturally to fair-sharing and equal-groups problems, where there is some group of physical objects being "dealt out." The array tool is best for area and (of course) array problems. Thus, the implementation of machine analysis based on math knowledge focuses primarily on these particular types of problems.

The array tool is also useful for teaching students to break a difficult problem into smaller parts. By adding horizontal and vertical dividers to an array, students can divide the array into a number of smaller rectangular regions. The area of the full array can then be found by calculating the area of each small region and adding them all together. In classroom trials, students have worked on multiplying two- and three-digit numbers using the array tool, whereas it simply would not be practical to add that many stamps to a page. Therefore, the semantic interpretation of the array tool needs to take into account the different strategies students might use to solve problems involving larger numbers, particularly by reducing them to a set of simpler problems.

To achieve this goal of analyzing student work, it was necessary to represent math domain knowledge and reason about that knowledge. We created a representation called a "page definition" that encapsulates the math problem associated with a given page. The page definition currently includes the type of problem (see Section 2.2), the product relation featured in the problem, and which values in that product relation are given in the problem.

Once the students' work has been analyzed by CLP, the results of this reasoning are made available to the teacher using CLP's tagging structure. “Tags” are objects that may be added to a page; each tag has a type and one or more values. The analysis module automatically creates tags and applies them tags to a student's submitted page based on the page definition and the mathematical interpretation of the page objects added by that student. After receiving pages from multiple students, the teacher can select any of the available tag types and view a list of all submissions sorted by the value of that tag, enabling him or her to quickly identify students who used a particular strategy or made a particular common error.

This project was developed with the following primary design goals in mind:

- The creation of page definitions needs to be as simple and intuitive as possible, so that page authors need not know anything about the inner workings of CLP.
- At the other end of the process, the end result of the analysis step needs to provide useful and immediate feedback about the students' mathematical representations. In particular, it is assumed that the teacher is interested not only in correct vs. incorrect answers, but also in the methods by which each student arrived at a final solution, whatever it might be.
- Finally, the page definition and tagging structures need to be sufficiently flexible to accommodate new types of problems or problem-solving strategies that teachers may request in the future.

4 Implementation

4.1 CLP's Representation

As mentioned earlier, CLP is organized around the notion of pages, each of which contains a particular math problem. The problem's math domain knowledge is stored on the page in the structured object called the "page definition."

A page definition comprises one or more product relations, and optionally some number of mathematical constraints. A product relation contains three values: two factors and their product. These values are ordinarily integers, but may also be variables (represented as strings). A product relation is also assigned a relation type, such as "equal groups" or "area." This distinction is used in the machine interpretation step, and also determines how the integers are labeled for the page author or teacher. In an equal-groups problem, the factors are "items per group" and "number of groups" and the product is "total number of items"; in an area problem, the factors are "height" and "width" and the product is "area." Finally, each value in the product relation can be marked as "given" to indicate that the value is given as part of the problem. In this paper, non-given values (or "unknowns") in page definitions are indicated with square brackets.

Consider the following problem from a hypothetical workbook page: "The whiteboard in Ms. Johnson's classroom is 4 feet tall, and has a total area of 24 square feet. How many feet wide is the whiteboard?" This page's page definition would consist of a single product relation: "Area: [6] (width) x 4 (height) = 24 (area)". The brackets around the number "6" indicate that 6 is not one of the values given by the problem.

The addition of constraints would allow the page author to create problems that are more complex or have multiple possible solutions. A constraint would consist of an equation or inequality that uses one or more of the variables defined in the page's product relations. See Section 7.1 for additional details.

4.2 CLP's Reasoning

This section contains a list of the tags that can be automatically generated and applied to a page based on the interpretation of its page objects in the context of the page's mathematical content, which is stored in its page definition. Tags are listed as they appear in the teacher's sorting UI, and each entry contains a description of that tag type's possible values and how the analysis routine determines which value(s) to apply.

4.2.1 General-Purpose Tags

Correctness. This tag is applied after this project's analysis routine compares the page objects on a submitted notebook page to the attached page definition, and determines whether the student's work is an accurate mathematical representation of the product relation in the problem. This tag is given a value of “Correct” if there is a match, and one of several “Error” tag values otherwise. The value “Error: Swapped Factors” is applied in cases where the student creates page objects that depict a product relation that has the correct factors and product, but doesn't properly account for the difference between the two factors in the context of a problem. For example, this tag value will be applied if the page definition of a page is “Equal groups: [5] (number of groups) x [3] (items per group) = 15 (total items)”, and the student instead places 3 groups of 5 stamps each. The student may still come to the correct conclusion that there are 15 items in total, but this tag can alert the teacher that perhaps the student did not completely understand what the story problem was asking. The “Error: Misused Givens” value is applied to this tag if the student creates a representation that uses the given values in the problem, but does not relate them to each other in the expected way—for instance, if the student is asked to divide 4 by 2 and instead uses stamps or an array to show that 4 * 2 = 8. A page that does not contain a correct representation of the problem but does not match either of the errors above is given a Representation Correctness tag with a value of “Error: Other”.

Math Tool Used. This tag signifies what type(s) of page objects the student added to the page to show their work. The possible values are “array” and “stamps”. If a page contains both stamps and arrays, this tag will be given both values and applied to the page.

4.2.2 Stamp Tags

Parts Per Stamp. The value of this tag is an integer corresponding to the number of parts in each stamp, as entered by the student while creating the stamp.

Grouping Type. This tag type exists mainly for troubleshooting the syntactic interpretation step and is hidden from the teacher's view by default. The possible tag values are the different methods the group-detection algorithm uses to infer the student's intended distribution of grouped stamps: “Basic Grouping”, “Ink Grouping”, “Distance Grouping”, “Container Grouping”, and “Container Distance Grouping”.

4.2.3 Array Tags

X-Axis Strategy and **Y-Axis Strategy**. These tags show the problem-solving strategy the student used to subdivide the array into smaller pieces. The common strategies currently represented as tag values are “place value,” “10's,” and “half.” (See figures 4-1, 4-2, and 4-3 for examples of these strategies.) A student using the “place value” strategy divides the rows (or columns) of the array into one region for each digit in the height (or width) of the array. For example, after placing an array with 423 columns, the student would insert vertical dividers to separate the array into regions of 400, 20, and 3 columns, respectively. The “10's” strategy involves splitting the array into smaller regions whose dimensions are powers of 10 (i.e., 1, 10, 100, etc.). Using this strategy, a student might split an array with 47 columns into five regions: four with 10 columns and one with 7 columns. The “half” strategy entails simply splitting an axis of the array into two regions of equal size. In detecting these strategies, the algorithm does not take into account the particular ordering of regions along the axis. If the student uses an unrecognized strategy to split the array, this tag is applied with a value of “other”. If the student does not add any dividers along an axis, this tag is applied with a value of “no dividers”.

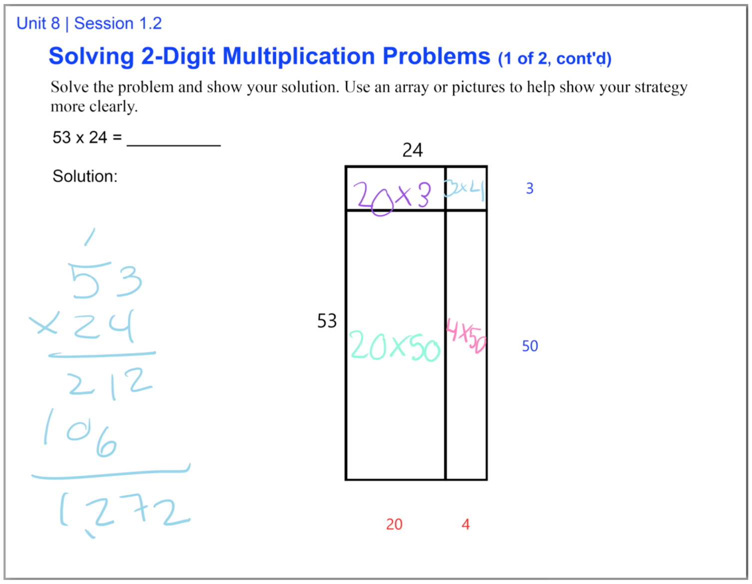

Figure 4-1: An example of the place value strategy.

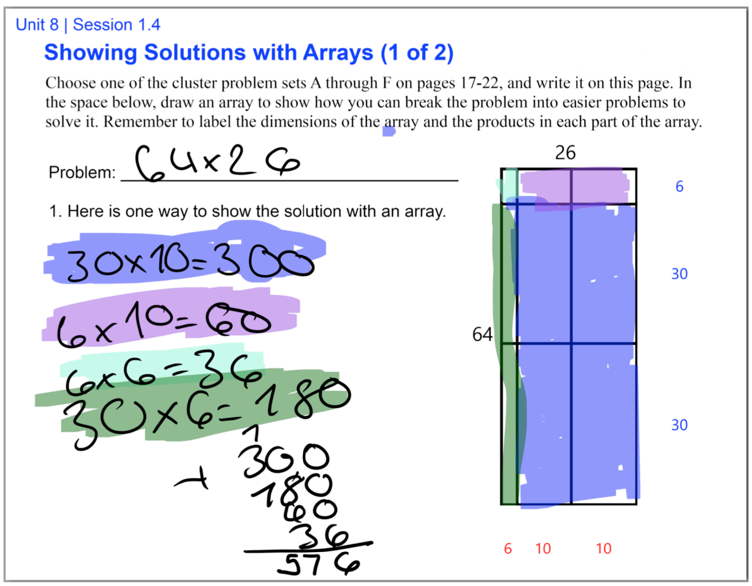

Figure 4-2: This student used the 10's strategy to divide the array's 26 columns into three regions containing 6, 10, and 10 columns.

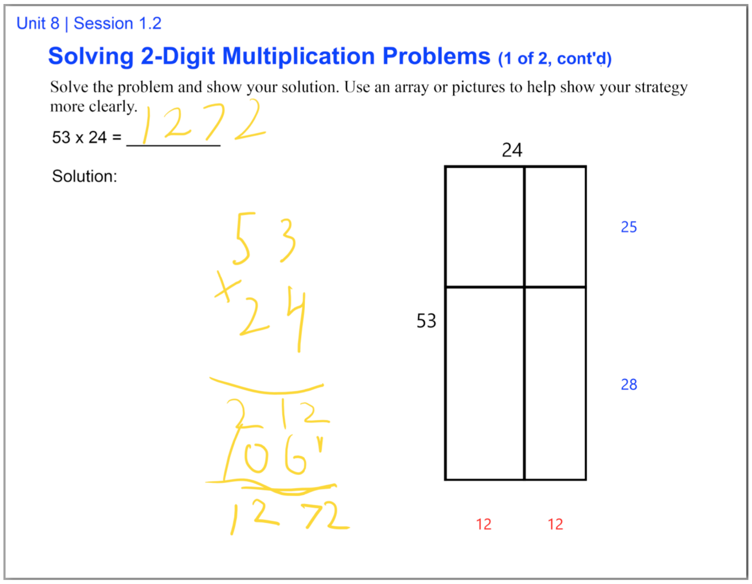

Figure 4-3: This student used a vertical divider to split the array's 24 columns into two groups of 2 columns.

Array Divider Correctness. This tag shows whether the student entered values for the height and width of the smaller array regions that add up to the height and width of the full array, respectively. Possible values are “Correct”, “Incorrect”, and “Unfinished” (which is applied if the student did not specify values for all array regions).

Regions Formed by Horizontal Dividers and **Regions Formed by Vertical Dividers**. The value of each of these tags is an integer, determined by the number of smaller regions formed along one axis of the array using dividers.

Orientation. This tag is given a value of “First factor is width” or “First factor is height”, where the “first factor” is the first one listed in the page definition. If the student's array does not match the factors in the page definition in either order, this tag is added with the value “unknown”.

Overall, the strategy has been to minimize the total number of tag types, in keeping with the design goal of keeping the teacher UI as simple as possible. Extensions to this tagging framework should focus on adding new values to existing tag types where possible. For instance, “Swapped Factors” and “Misused Givens” were originally distinct tag types, but were consolidated into the “Representation Correctness” tag type as values instead.

5 Discussion

One interesting finding resulting from this work that could be further explored is the interaction between types of problems and the various mathematical representations used to solve them. For instance, there is a meaningful difference between three groups of four stamps and four groups of three stamps, and pages using those two strategies should receive different tags to allow the teacher to quickly differentiate between them. However, whether this difference affects the *correctness* of a solution depends on the type of problem. In an equal-grouping problem ("There are 3 bags containing 4 lollipops each; how many lollipops are there in all?") only one of those groupings is correct, whereas in a generic multiplication problem without units ("What is 4 times 3?"), both solutions are equally valid.

A comprehensive math curriculum should include examples of all the types of multiplication and division problems listed in Section 2.2, but there are several types of problems that are not well-served by any of the tools currently available to students. For instance, it is not especially intuitive to represent a rate problem using either stamps or an array. Fully exploring these types of problems will require new modes of input for students, along with new methods for analyzing their work. One new tool currently under development—the number line—may prove useful for rate, cost, unit conversion, and scaling problems.

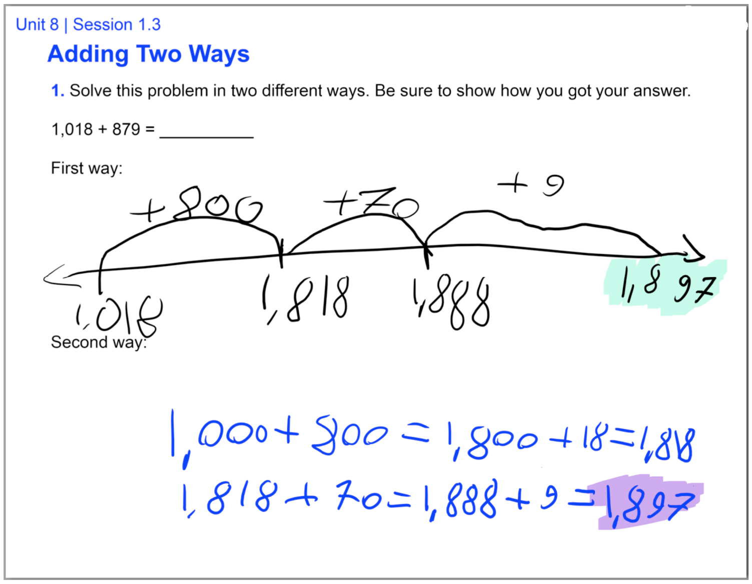

Figure 5-1: A student solves an addition problem using a number line drawn with the ink tool. The number line tool will allow students to create a similar representation in a structured way that is more amenable to machine interpretation and analysis.

Cartesian product problems involve calculating the number of unique combinations of items from multiple sets of objects. For instance, how many different ways can one combine one of three flavors of ice cream with one of five possible toppings? The existing stamp tool might help students reason about this sort of problem, but to automatically analyze their work we would need to develop new methods of syntactic and semantic interpretation because the students would be using the stamps in a very different way.

We have experimented with using handwriting recognition in order to interpret students' written work. Our work to date suggests that the built-in Microsoft recognizer that we have been using does not have high enough recognition rates to be used in upper elementary school. (Our investigation of this issue is ongoing at present.) A more robust handwriting recognition system would enable CLP to recognize new ways in which students might illustrate their mathematical reasoning. For instance, the page author could indicate a place for the student to write their final answer, or even an entire number sentence representing the problem—which is of course a representation that is applicable in any type of multiplication or division problem. The analysis routine could also check for numbers and number sentences written inside the subsections of an array, to generate tags related to the student's ability to compute partial products.

6 Future Work

6.1 Constraints

As noted in Section 4.1, mathematical constraints were part of the design for page definitions but are not yet implemented. These constraints would be equations or inequalities using variables specified in the page definition's product relation(s).

Constraints can be used to construct a problem with multiple valid solutions. Consider an example problem: "Alice has 24 candies that she would like to put into bags, so that each bag has the same number of candies. She wants to use at least two bags, but no more than eight. Use stamps to show one possible way she might do this." The definition for this page would require one relation: "Equal groups: [**X**] (number of groups) x [**Y**] (items per group) = 24 (total items)", where **X** and **Y** are variables, and 24 is the only given value. The author would then add two constraints: "$\mathbf{X} \geq 2$" and "$\mathbf{X} \leq 8$". Thus, the author has created a problem with five possible solutions, without needing to find and input each solution separately.

Constraints can also be used to connect two separate product relations. Consider another problem: "Tricycles have 3 wheels and wagons have 4 wheels. Bob wants to buy some number of tricycles and wagons with a total of 26 wheels. What is one way he could do this?" This page definition would require one relation for each type of vehicle: "Equal groups: [**T**] (number of groups) x 3 (items per group) = [**X**] (total items)" and "Equal groups: [**W**] (number of groups) x 4 (items per group) = [**Y**] (total items)". A single constraint would then be added, to enforce the total number of wheels: "$\mathbf{X} + \mathbf{Y} = 26$".

In both of the above cases, the automated analysis step would then infer what integer value a given student used for each unknown variable and attach that information to that student's page as a tag. This would allow the teacher to quickly find examples of two different solutions and present them to the class.

6.2 Other Extensions

If there is sufficient need, entirely new components could be added to the page definition structure. For instance, some word problems contain “decoy numbers” that are not actually relevant to the problem the student must solve. The page definition structure could be expanded to include an option for decoy numbers. New analysis routines would be added to recognize when the student uses the decoy number to create an erroneous representation, along with new tags (or possibly just new error values for the “Representation Correctness” tag) to add to such pages.

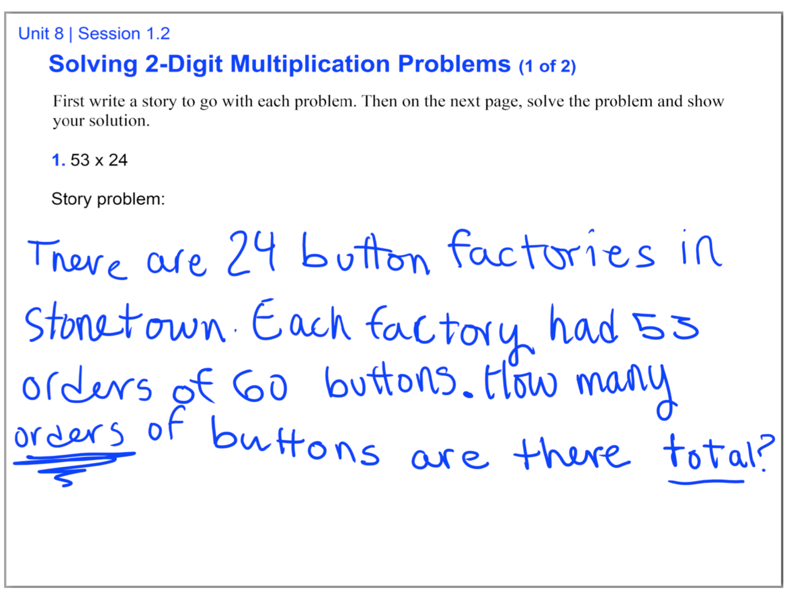

Figure 6-1: When asked to devise her own story problem to model a multiplication operation, this student included an extraneous number (60) as a potential trap for a careless reader.

Additional tag types can and probably will be added in the future, along with new values for existing tags; this project was designed with that need in mind. Adding the representation for new tags and values is straightforward, while the difficulty of implementing the reasoning to determine whether a tag is applicable depends on the tag itself.This process will require further field testing, to determine what additional metrics educators would like to use to sort and filter students' work. One new suggestion from early testing with educators is to include "Array Strategy" tag values that are based on the partial products created by the student's splitting of the array. For instance, the analysis tools could single out students who manage to split the multiplication problem into subproblems involving "friendly numbers"—that is, numbers that are easy to manipulate, such as multiples of 5 and 10.

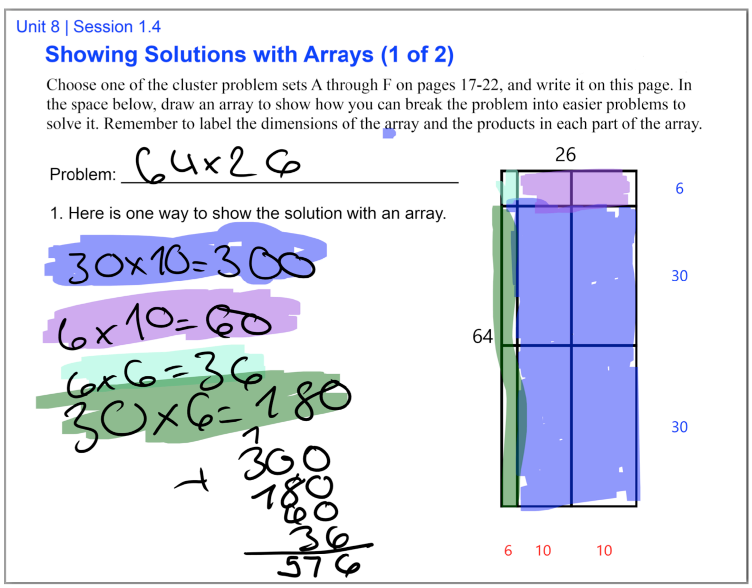

Figure 6-2: This student used horizontal dividers to split this array's 64 columns into three regions containing 6, 30, and 30 columns. Tags could be applied to this page to signify notable features of this strategy, which A) creates partial products that are nearly all multiples of 10, and B) creates multiple regions with the same dimensions, reducing the number of multiplication operations the student must perform. This student helpfully color-coded these same-sized regions, but failed to account for the multiple regions in the addition step and obtained an incorrect answer.

The analysis routines developed in this project could potentially be applied before the student even submits their work. For instance, the “Array Divider Correctness” tag could be used to alert students when they assign values to array subregions that do not add up to the number of rows or columns in the array. A teacher might want to enable such an option to help students avoid simple addition errors so that they can focus more on the multiplication lesson at hand. One downside of this approach is that it would force the notebook author to consider on a case-by-case basis what level of feedback is most appropriate for a given lesson.

The teacher UI could be further refined to create a more streamlined experience. The page definition dialog could have some internal validation to prevent the page author from populating a relation with incompatible values. The tag-sorting interface would also benefit from the ability to display a page below multiple tag values. For instance, a student might use different strategies to split the horizontal and vertical dimensions of an array. Their page will be assigned an “Array Strategy” tag with two values, but the page will appear only once in the teacher's sorted list, under one of those two values.

7 References

[1] Confrey, J., Nguyen, K. H., Lee, K., Panorkou, N., Corley, A. & Maloney, A. P. (2012). Turn-on common core math: Learning Trajectories for the K-8 Common Core State Standards for Mathematics. Retrieved from http://www.turnonccmath.net.

[2] Koile, K. and Rubin, A. Machine Interpretation of Students' Hand-drawn Mathematical Representations. *Workshop on the Impact of Pen and Touch Technology on Education* (WIPTTE) 2013.

[3] Maynard, E. "Using a Structured Vocabulary to Support Machine Understanding of Student Work," Master of Engineering Thesis, MIT, 2013.

[4] Rubin, A., Storeygard, J, and Koile, K. Supporting Special Needs Students in Drawing Mathematical Representations. *Workshop on the Impact of Pen and Touch Technology on Education* (WIPTTE) 2013.