Graspable Math: Towards Dynamic Algebra Notations that Support Learners Better than Paper

***Abstract*—Pen and paper remain the preferred tools for solving mathematical problems – despite the fact that paper derivations are very challenging to master and are limited by the static nature of paper. While traditional computer algebra systems are not geared towards the need of learners, dynamic algebra notation systems (DANS) that bring derivations into the digital space without sacrificing the user agency and directness that paper derivations provide have much to contribute to mathematics education. In this paper, we first analyze the strengths and weaknesses of pen and paper for doing algebraic derivations and discuss the potential benefits of DANS. We then present *Graspable Math*, our own implementation of such a system, discuss the interface design challenges solved in its development, and report on the encouraging outcomes of two user studies.**

Keywords—e-learning; dynamic notation systems; algebra; interactive mathematics; mathematics education

I. Introduction

Formal mathematical notations such as algebra present a basic conundrum for mathematics teaching and learning. On the one hand they are tremendously powerful in the hands of experts and gate careers and innovation in STEM. On the other hand, students mostly get frustration out of them and only 33% of 8th graders and 25% of 12th grade students in the United States reach proficiency in mathematics (NCES, 2015). Our research is focused on the role of technology and interfaces that mathematical reasoners use to work with algebra notation. For most people today, this interface is pen and paper. We believe there is great potential in designing e-learning tools with new interfaces that go beyond the capabilities of paper in supporting learning and teaching in mathematics. We are presenting one such tool, called *Graspable Math*, in this paper.

Many of our thinking and communication tools have been transformed by computers, including mathematic tools. We can neatly typeset equations with formula editors and systems like LaTeX, automatize long or complex derivations with powerful computer algebra systems (CAS) like Mathematica or Maple, and provide immediate feedback to students by checking the answers to their math homework with e-learning systems like ASSISTments, to name just a few. However, when it comes to learning how to manipulate algebraic expressions though, existing CAS are of limited use. Math users, from middle school students learning basic mathematical concepts to professional mathematicians discovering new theorems, often prefer traditional paper over these systems and they do so for good reasons.

A. Traditional Paper

User agency, flexibility, and directness are among the unique advantages of paper for mathematical derivations [1]. Paper maintains user agency in that at each point in transforming a mathematical expression, one chooses among different paths by deciding on the next step, promoting in-depth understanding and insights. Paper is flexible in that one can write or draw any form of representation in any level of detail and rigor, as well as easily annotate content with notes and supporting material. In addition, paper provides a direct user interface in that the mapping between one's actions (writing) and the results (scribbles on the paper) are directly apparent.

It is the simple and static nature of paper that leads to these advantages. However, the same simplicity does not provide a good match to the intrinsically complex nature of most mathematical domains and makes learning difficult. Because any expression can be written after any other, the user of paper is tasked with maintaining mathematical integrity by avoiding syntactic and conceptual errors. To avoid these errors on paper is hard, since one has to simultaneously accomplish a number of challenging tasks such as copying information reliably from one place to another without making transcription errors, correctly applying mathematical transformations to appropriate subexpressions of a potentially complex and deeply structured expression, as well as finding and correcting errors through repeated checking. These low-level skills must be highly trained before students can focus on high-level mathematical aspects of algebra problems.

Another less obvious problem with paper is that it does not capture the fundamentally dynamic nature of algebraic transformations. Paper is poorly matched to people's conception of proofs, which often involve continuous, visuospatial transformations of complex symbol structures [2]–[4]; these transformations are difficult to express in a static medium like paper. Consider the short derivation

$$\begin{aligned} 3(x+y) &= 3y \\ 3x+3y &= 3y \\ 3x &= 0. \end{aligned}$$

One could simply consider the application of steps as rewriting rules: the assertion that $3(x+y) = 3y$ licenses the assertion that $3x + 3y = 3y$. However, people typically think of this much more narratively: we 'move' the 3 into the parenthesis; the $3y$ on the left and right side of the equal 'cancel'. Understanding proofs as narratives requires a user of paper to map the structures in one line into those of another, re-identifying common elements.

B. Dynamic Algebra Notation Systems

How can e-learning tools retain the user agency, flexibility, and directness that make paper derivations so useful, while providing more structure and feedback to support learners?

We believe that a particularly promising approach is to use dynamic computer algebra systems that allow users to perform step-by-step derivations and allow interacting directly with the mathematical expression that is transformed.

There are several recently developed systems that focus direct manipulation interfaces for creating mathematical proofs, in particular SetPad, Hands-on Math, MST, and MathPad^2 [5]–[8]. All these systems allow the user to hand-write formulas on the screen and to manipulate them using gestures, while (with the exception of MST) automatically ensuring mathematical consistency. Hands-on Math and MathPad^2 seek to provide an intuitive interface for existing classical computer algebra systems and allow users to connect several representations with each other. SetPad, Hands-on-Math and MST facilitate step-by-step derivations through gestures in their domains of set theory, algebra and algorithmic proofs, respectively. Other systems, such as the iPad applications DragonBox, AlgebraTouch, and FromHere2There [9]–[12], use the interface metaphor of algebraic terms as physical objects. Terms can be picked up and moved by the user to trigger mathematical transformations and smoothly move to their new locations after the transformation.

Moving derivations into the digital space allows going beyond what is possible with paper in many ways. For example, derivations can be automatically recorded as students create them and can then be easily shared with other students or teachers. Dynamic algebra notation systems can provide immediate feedback on potential errors and support users in applying individual transformations to create a save space in which even novices can explore and play with algebraic expressions. Additionally, digital derivations can be automatically annotated to highlight connections between lines and support insights.

In order to further explore the potential of dynamic algebra notations, we decided to develop our own system. One reason behind this is that most of the existing systems were limited to a particular device, a narrow domain, or are not readily available to the public. We wanted a system that is easy to access for learners, teachers, and researchers, can be used across different devices, and can be integrated into existing e-learning tools. Another reason was that, while each of the existing systems has its particular strengths, such as the smooth interactions of DragonBox, or the expressiveness and power of MST and Hands-on-Math, we felt there was no system that combined them in a way to be competitive to pen and paper for learners.

II. The Graspable Math System

Graspable Math is a web-based dynamic algebra notation system with a focus on a consistent, efficient and powerful user interface for manipulating algebraic expressions. Currently, GM covers the contents of middle school algebra and supports direct, in-place interactions with the algebraic terms via touch or mouse gestures. GM currently supports real numbers, variables, signed expressions, brackets, fractions and nested fractions, equations, inequalities, power expressions, absolute values, as well as a big library of transformations (actions) on those expression types.

The development of GM was guided by the concept of direct manipulation interfaces [13] and their advantages in terms of ease of learning and use. We focused on developing a consistent gesture language in which individual math transformations are triggered by dragging the involved terms to the place they should end up after the intended transformation. For example, distribution is triggered by dragging a factor into a sum. GM consistently uses the metaphor of physical objects for algebraic chunks in its interface. It visualizes transformations as smooth transitions between states, providing visual continuity of all terms, which allows users to visually track how terms move while an expression is transformed into another. A considerable body of research emphasizes the importance of such perceptual aspects of learning setups even in abstract domains like algebra [4], [14].

To support insights in the connections between terms, GM allows inspection of derivations by recording and organizing them in lines and using colored highlighting to show how a term transforms across the lines. The GM workspace combines these features with the affordances of a classical whiteboard, so that users can make flexible annotations. Finally, GM is implemented as a web application that can be used on any device with a modern web browser and can be integrated into existing web-based e-learning tools and platforms.

We describe some of GM's central features and design decisions in more detail below. Many of those decisions were made in response to user studies.

A. Gestures and Animations

All gestures for transforming an expression act directly on the terms that are involved in the transformation. Whenever the user picks up a term, GM creates a set of target areas, one for each action that can be performed in the given context. When the term is dragged on top of one of the target areas, the respective action is triggered; the specific trajectory to the target area is irrelevant. This design keeps the gesture language that users have to learn simple – in most cases a gesture simply involves dragging a term to the place it should move in the intended transformation. Figure 1 shows the gesture used for distributing a factor into a sum. When the user picks up the factor and drags it towards the sum, the system automatically performs the distribution once the factor overlaps with the sum, which is the target area of the distribution action. All terms transition smoothly to their new locations. The user can now either drop the factors to finish the interaction or keep moving the factors out of the sum to revert the distribution.

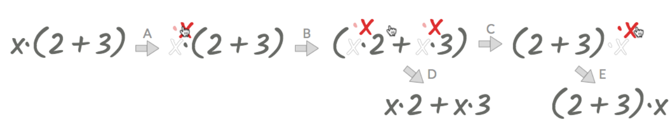

Fig. 1: Distribution in GM. The user picks up the x (A) and drags it into the brackets which triggers the distribution (B). The expression is updated in-place to reflect the result of the distribution. The user can now either finish the gesture now (D) or after dragging the selected $x$es out of the sum to factor them (C,E).

There are two types of interactions for performing actions on algebra expressions in GM, dragging and tapping. Some of the actions that can be triggered through tapping are adding numbers, fractions and like terms, multiplying numbers and variables, simplifying signs, dividing numbers, raising sums or numbers to a power, or swapping the sides of an equation. Dragging can be used to commute terms, move terms across the equal sign, factor and distribute with sums, products, and power expressions, move terms in and out of fractions, cancel fractions, as well as to perform substitutions. Additionally, dragging numerical addends or factors on top of each other will combine them. We refer to this gesture as *number smooshing* or simply *smooshing* (see Figure 2).

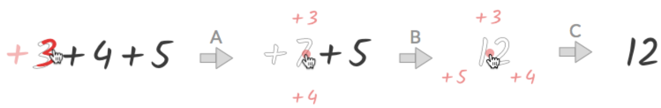

Fig. 2: Number smooshing in GM. The user has picked up the left-most addend "3" and drags it on top of the "4" (A). This combines both addends, showing the result as a *shadow*, while the addends are still visible in a smaller font and arranged on a circle. Dragging on top of further addends combines these, too (B). The smooshing interaction allows to perform multiple operations in a single gesture, instead of requiring several clicks or taps (C).

A powerful feature of Graspable Math is action chaining. This allows the user to trigger a sequence of transformations through a single, uninterrupted gesture. Since all gestures and transformations provide immediate, in-place visual feedback, the user can continuously drag terms to perform multiple steps without lifting the finger, allowing for efficient interactions. See Figure 3 for an example. This feature was directly motivated by early user surveys: In previous versions, executing simple operations such as addition was done by tapping the operation sign (as in Algebra Touch), and in these versions a single action was performed with each mouse click. Early users were frustrated that they could not take 'shortcuts' (for instance, combining $2x+5+3x+7$ by simultaneously regrouping and adding like terms), that they seemed to calculate internally during traditional problem solving. As a result, participants uniformly felt that these instantiations of GM had the opposite of our intended effect, relative to paper, shifting focus from high-level tactical and structural considerations and onto low-level rearrangements. In the words of one of these participants, it was a 'click fest'. It was in response to these concerns that we introduced smooshing and action chaining.

We support multi-touch, single-touch, or mouse interactions in GM. One reason behind this that although multi-touch gestures are very expressive, they are not available on many widely used devices. Another reason is that from our experience, people often prefer single-touch gestures over multi-touch gestures, especially if the exact positioning of the fingers is required. Therefore, for each multi-touch interaction, there is also an alternative single-touch gesture (which may involve a keyboard). For example, picking up several terms at once can be done by placing two fingers around them, by shift-clicking to select the terms, or by pressing the space bar after selecting a term.

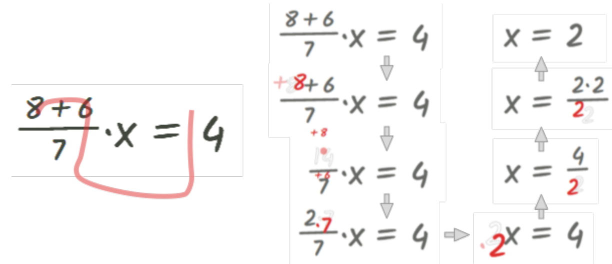

Fig. 3: Action chaining in GM. The equation on the left can be solved with a single, continuous mouse gesture, which is shown in red. At the right are the states the the expression goes through while the user performs the gesture. All changes happen in-place during the gesture and unpack into separate steps after the interaction is finished. Smart reselection of terms after each action and having most actions be triggerable through dragging allows for long action chains like this.

B. Workspace

The workspace acts like a whiteboard and holds derivations and drawings the user made. Users can start a new derivation by bringing up an on-screen keyboard and entering the first line of the derivation. Once a derivation is created, it can be moved around freely and manipulated on the workspace. The possibility to freely draw or write on the workspace allows users to annotate algebraic expressions (see Figure 4). Besides the students themselves making annotations, teachers and tutors can use this feature to provide written feedback to students in an online course.

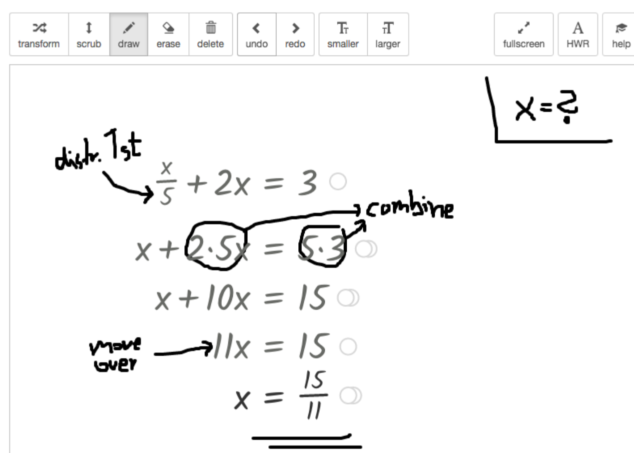

Fig. 4: GM Workspace. The user can create, manipulate and rearrange algebraic derivations on the workspace, as well as freely draw to annotate such derivations.

C. Mathematical Narratives

GM automatically records the lines of a derivation list and the terms in them are connected to each other through the transformations that were performed. This allows GM to visualize how terms map onto each other. The user can switch to "inspect mode" and tap on any term in a derivation to see its path through the derivation (see Figure 5 for an example).

All terms in previous lines that directly influenced and all terms in following lines that resulted from the selected term are highlighted. This feature aids the user in getting an overview of the big "themes" or "motives" in a mathematical narrative.

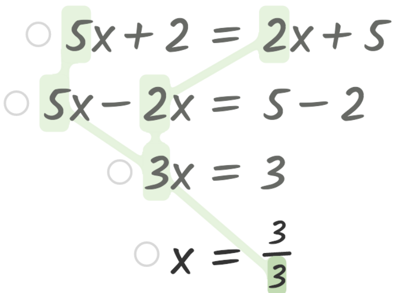

Fig. 5: Term Mappings in GM. This derivation of the x-coordinate of an intersection of two lines is annotated with a visualization of which terms turned into the selected 3 in the denominator. The user can select any term to see a visualization of how this term 'moved' through the derivation.

To tell a consistent mathematical story, it is often necessary to combine some of the smaller steps that were taken into meaningful packages. In Graspable Math, the user can do this by picking up lines and moving them up on top of previous lines (see Figure 6). Pulling a line down reveals the hidden lines below it. This interaction is modeled after the metaphor of a set of connected paper cards – one for each derivation line – that can be stacked on top of each other. The visualization of term mappings works across stacked lines.

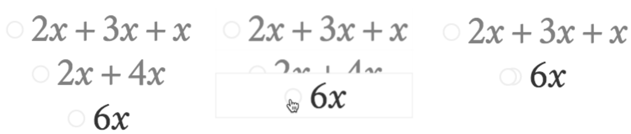

Fig. 6: Stacking lines in a derivation. The user can collapse several lines in a derivation by dragging one line on top of the lines above it. Pulling the line down reverts the collapsing.

D. Design Challenges and Solutions

1) In-Place Manipulation: Direct and in-place manipulation of the math expression via dragging of terms provides many advantages and sets GM apart from most of the systems we reviewed. Supporting in-place manipulation comes, however, with a set of challenges. First, changing the very expression a user is currently working on can be disorienting, and second, the result of an interaction might be hard to predict for the user when the same expression is used to reflect the updated state and as the object to trigger changes on.

In Graspable Math, we tackle the former point by using smooth animations that provide continuous transitions of the initial terms and their positions to the new ones. Terms don't disappear and reappear, but instead move from the original to their final positions. We provide an additional measure to facilitate visual understanding of transformations: GM delays some of the automatic simplifications like removing brackets after distribution to after the user finished the interaction (see step D in Figure 1). This allows us to temporarily separate part of the complexity of the transition. Additionally, it makes undoing of an action by dragging terms back to their old position easier, as fewer structural changes are performed before the user commits to the action.

In order to help the user see what an expression will be when the user finished the current gesture, we show a gray outline of the dragged term at the position it will move to when it is dropped (see Figure 1). We refer to this as *term shadow* or short *shadow*.

2) Efficient Action Sequences: When people do algebra on paper, they typically take mental shortcuts or combine several steps before writing down the next line. One way to make an interactive algebra system scale with the skills of a user and provide the same feeling of flow without sacrificing the step-by-step control is to make the process of triggering a series of transformations as smooth and effortless as possible. One central concept in GM is the chaining of transformations within a single dragging interaction. Since transformations are performed in-place and during the dragging, it is often possible to continue to drag after a completed transformation without interrupting the gesture. The challenges in designing such an action chaining system is that the system has to, first, make intelligent decisions which terms to select for dragging after a transformation was done. Second, most actions must be triggerable through dragging, and third, the interaction might suffer from involuntary triggering of further actions when the terms in an expression – and the associated target boxed – move after a transformation.

In GM, we use the mapping of old to new terms that is defined by each transformation to select the terms that are being dragged in after a transformation was applied. For example, after moving the x in $x(2+3)$ into the brackets, the new state becomes $(2x+3x)$ with the the two x'es being automatically selected for further dragging. (See Figure 1).

In order to allow for long action chains, most actions including canceling in fractions, rewriting equations and factoring can be triggered by dragging. We recently added the functionality of adding and multiplying numbers via dragging them on top of each other – or "smooshing" them. Figure 3 provides an example of an equation that can be solved with a single, continuous dragging gesture.

Finally, avoiding the unintended triggering of follow up actions right after an action was done is a tricky problem. This is because the underlying structure and positions of all terms can change dramatically between steps and with them the target areas that trigger the actions. If after a transformation, all terms in an expression are moved to their correct new locations and the position of the dragged terms is kept fully under user control, it is possible that the dragged terms overlap with a new target area, triggering the next action immediately and without user intent.

There are at least three approaches to resolve this unintended action triggering, yet all of them have different trade-offs. One option is to carefully choose where to place target boxes and to add situation-specific rules to avoid the immediate triggering. We tried this approach with mixed success, and found that it often breaks down in novel situations or when we added new gestures or capabilities. A second option and general solution is to reposition all terms relative to the dragged terms, instead of moving them to their actual new positions after a transformation. Since it is typically the horizontal rearrangement of terms that causes triggering problems, the application of relative repositioning can be limited to the horizontal positions. The downside to this solution is that the horizontal position of the whole math expression now slightly changes during a gesture and moves back to its correct absolute position only at the end of each interaction. For most use-cases this does not have a negative impact though, and we are currently using this approach with good success. A third option is to move all terms to their correct new positions, and to instead adjust the position of the dragged terms. We found that this option interferes with the user experience of holding and moving terms, since dragged terms might not follow the users movements at the moment a math action is triggered.

3) Visualizing Term Connections: Visualizing the path an algebraic term takes through a derivation requires knowledge about the connection of terms between every pair of consecutive lines in the derivation. In GM, we define the mapping of each previous term onto a set of new terms for each of the available transformations. A term might be mapped onto several, as is the case in distribution, it might be mapped to a single term as is the case in commuting terms and it might be mapped onto an empty set, as is the case when removing optional brackets.

When automatically combining several transformations or when displaying term mappings across hidden derivation lines, we compose the term mappings of several actions to arrive at a mapping that spans several transformation steps.

4) Learning Curve: With any new system, users require training and practice to use it efficiently. To help new users learn how to use GM quickly, we designed an interactive gesture tutorial that walks the user through the gestures for transforming expressions. For each available gesture, the tutorial combines short videos of how to apply the gesture with an area that lets the user perform the gesture on the same example as in the video. Once the video has finished, we provide further assistance by making the term that the user should pick up wiggle. It currently takes about ten minutes to work through the tutorial. Although of course it takes longer to fully acclimate to the interface, in our experience users have little trouble remembering and using the basic gestures after this short tutorial. The tutorial is available at http://graspablemath.com/tutorial and works best in Chrome.

III. User Studies

User studies played a formative role in the iterative development of GM. We describe two studies in which we collected feedback to guide development and tested the feasibility of the system as an alternative to paper derivations. For these studies, we emphasized the procedure of solving linear equations in one variable. We selected this task because it is a standard problem in high school mathematics and science, and is reflective of many routine tasks outside of these domains. Furthermore, our belief is that any replacement for paper must be at least as convenient as paper. Therefore, fluid, accurate equation solving forms a reasonable baseline domain on which to evaluate a computer algebra system.

A. Pilot Study

We first report the results of a small pilot study of the usability of the system, conducted in early April, 2015, with a heterogeneous population of 14 individuals. Participants included 11 members of the psychological research lab of the second author (1 faculty member, 3 post-docs, 1 graduate student, 2 former undergraduates and 4 current undergraduates). Of these, 4 had substantial experience with the system, either as users or as programmers; the remaining 7 had minimal exposure to the system.

1) Material: Equations A1 through B3 present the full set of problems used in the main pilot activity. These problems were designed to be reasonably complex and challenging, and to explore several features of Graspable Math, without presenting so many features that new users would be completely overwhelmed. The subjects had the task to solve the following two sets of equations using pen & paper for one half and GM for the other.

$$\tfrac{2+4x}{5} = \tfrac{8x}{5} + 6x \tag{A1}$$

$$5y + \tfrac{3z}{z} + 6y = \tfrac{2+4y}{5} + \tfrac{3y+7}{5} \tag{A2}$$

$$\tfrac{5(3+2x)}{8x} + 7 = \tfrac{16}{4} + 9 \tag{A3}$$

$$\tfrac{7b}{3} + \tfrac{6}{5} = b + \tfrac{4b}{8} \tag{B1}$$

$$4x + \tfrac{3y}{y} - 4x = 4x + \tfrac{5x+2}{4} + \tfrac{12x+4}{5} \tag{B2}$$

$$\tfrac{6(4x+2)}{2x} + 5 = \tfrac{20}{5} + 10 \tag{B3}$$

2) Procedure: The pilot study was intended to explore usability of the system for users with varying levels of expertise. We recognized that, as with any new interface, users would require instruction and practice in GM. Therefore, the pilot study began with a brief (roughly 15 minutes) instructional session on basic interactions. During this session, an experimenter showed each of several basic actions to the subject, first through a prepared video illustrating a single action, then by observing the subject attempt the action themselves. The actions illustrated were commutativity, moving terms across equations, distribution and factoring, selecting multiple terms, manipulating fractions, executing operations by smooshing and tapping, and chaining actions together.

After basic interactions, the experimenter led the participant through the solution of three complex problems, comparable to those in the main experiment. The goal of this training was to help the user understand features of the system in context, to practice recall of basic operations, and to remind subjects of the underlying mathematics and the strategies involved in solving equations.

After this training, participants solved problems A1 through B3. All participants solved set A before set B. However, for half the participants, set A was solved using GM and set B was solved with paper and pencil; for the other participants, set A was solved on paper and set B with GM. These problems were timed, and evaluated for successful solutions. Participants were encouraged to ask for help using the GM system during this time, but no help was provided on the appropriate mathematical strategy. Participants were also encouraged to discuss experiences, problems, frustrations, or observations during the study. This primarily occurred during use of GM, naturally, which may have increased the time and working memory load associated with the system.

After completing all six problems, participants were given a short questionnaire that asked about their thoughts and feelings on the system. They were also asked to report whether they thought they would prefer, in the future, solving problems like this in GM or on paper. In addition, verbal interviews were conducted in which participants were encouraged to reflect on their experiences, and describe strengths and limitations of the system.

3) Quantitative Results: As expected, participants made many errors in the lengthy derivations. On paper, participants successfully completed 57% of the problems. Although this may seem low, these problems were fairly complicated, requiring a large number of steps. Also, participants were not heavily motivated to check their answers (though many did at least some checking); undergraduate participants who lacked advanced mathematical training struggled particularly with these problems. As predicted, nearly all errors among all participants were either transcription errors, sign errors, or malformed transformations. In GM, 100% of problems were solved correctly, reflecting the fact that GM prevents trivial errors, and that subjects had some understanding of high level strategy. It should be noted that this 100% accuracy was reached despite the fact that subjects were allowed to give up on problems without completing them and that bad solution strategies in GM can lead to increasingly complex expressions. In previous iterations of pilot tests, accuracy with prototype versions of the system has been well below 100%. Solution strategies were broadly similar inside GM and on paper, suggesting that we are achieving our goal of creating a digital environment which supports algebraic reasoning by helping skilled users avoid simple errors.

The problems came in pairs (e.g., A1 and B1). A participant solved one of the problems in the pair on paper, and the other in GM. Response times were included only for problems in which a participant got both problems in the pair correct. Participants solved problems accurately at an average rate of 85 seconds per problem (range: 20–320 seconds). Participants were slightly faster using GM (Mean = 72s) than on paper (Mean = 98s), but the difference was not significant by a standard within-participants t-test ($t(10) = 0.85$, $p \sim 0.4$). We regard this non-difference as an achievement, though by no means an endpoint. Remember that people have had extensive practice using paper, while for most participants our system was quite new. Previous iterations of the system had, under similar testing conditions, been substantially slower to use than paper. It may seem intuitive that the calculator functionality of GM was responsible for its ease of use, but this does not match our observation. In part this is because the problems did not involve very difficult calculations, meaning that most participants could easily do the required arithmetic in their heads.

B. *Qualitative Responses*

Overall, participants reported satisfaction with the general gestural interface. Most gestures were regarded as natural and easy to remember. At the same time, they reported some difficulty actually implementing these gestures, in particular for gestures related to fractions. Many of the same participants struggled to manipulate fractions on paper, so some of their experienced difficulty may have come from the system violating their erroneous expectations. Another difficulty lay in the need to precisely place terms in desired locations. Finally, participants expected some gestures and transformations to work which we had not considered in our implementation. Based on this feedback we adjusted the target areas for fraction actions to make them easier to trigger and added several new fraction actions.

A second observation was that while action chaining clearly improved experiences of fluidity, it had unintended side effects. In the version of GM used in this study, action chaining used the first repositioning method described earlier, moving the equation to its new position while keeping the dragged terms at their place. This meant that often after one action the mouse would be positioned over a new object, and so a new, unintended action would trigger automatically. After the study, we changed the repositioning strategy and were able to completely avoid this unintended side effect.

One aspect of the system that surprised participants was that some complex actions could lead to large transformations. For instance, moving the $x+1$ in $\frac{4}{x+1}+2+3y=9$ across the equals sign triggered a shortcut in which $x+1$ is multiplied and distributed to both sides, yielding $4+2(x+1)+3y(x+1)+9(x+1)$. The secondary $(x+1)$ factors on the left, from the user's perspective, silently appear for no clear visual reason. This kind of action confused many users, while satisfying the strongest users. In part, this can be addressed by providing better visual clues to indicate where the additional terms come from. More generally, it suggests the need to adjust to the needs of a user.

A common frustration of users were the 'alt' and 'shift' methods of picking up large chunks. In an expression like $5x = 2x + 1$, participants would frequently intend to pick up the $2x$ by selecting either the 2 or the x; this would only select one term. Participants wanted a system that could better read their intentions, selecting just the elements they wanted. In response to this frustration, we implemented a "rubber band selection mechanism" that allows users to select additional terms by dragging the currently selected term further away. Nevertheless, selecting groups of terms remains one of the aspects of GM novice users struggle with.

Among the aspects participants liked about the system, they particularly emphasized the value of 'shadows' indicating the final state of the current transformation, and the general fluidity of the transformations. Most participants reported that they would want to use the system again, and that they preferred it over pen and paper for the purpose of solving complex equations.

On the basis of the pilot study, several bugs were fixed, several new gestures were added and existing ones were made more consistent and coherent. In our second user study which we describe next, we brought this improved but largely similar system into a classroom of middle school students. Although the changes were minor, we expected a substantial change in the user experience due to the increased fluidity and reliability of the next iteration of the system.

C. Middle School Study

One of our primary goals is understanding how dynamic algebra systems like GM can be leveraged in educational settings. We brought the revised system to a group of 48 middle school students participating in a summer math/science enrichment program at a small liberal arts college in a mid-Atlantic city. Participants had no prior experience with our system, in any iteration. Students were drawn from a number of urban middle schools, and were entering either 7th ($n = 23$) or 8th ($n = 25$) grade. Students applied for entry into the program, and all students were required to be in the top quarter of their regular-year classes in order to participate. Prior to participation in the study, participants had about 1 month of daily exposure to mathematical operations as part of the program, including (for the 8th grade group only) a particular focus on linear equations. Throughout the summer enrichment program, students were divided by grade into separate classrooms.

The students participated in a full-day intervention using GM, beginning with a short tutorial and algebra warm-up using both paper and pencil and GM. After the warm-up, students worked in a modified version of the program combining equations and linear graphs; they then completed a final round of the warm-up activity. Because the algebraic manipulations component is the focus of this paper, we report the results of the algebra warm-up portion of the activity only.

1) Materials: The warm-up activity was modeled after the pilot study, with modifications to make it successful in a classroom context. First, each student was given a laptop with an instance of GM, along with a random identification number (which permitted anonymous participation). Then, each student followed along as an experimenter illustrated the use of the program by stepping through a slightly revised version of the tutorial. The experimenter presented on the overhead, and the students followed along on their computers. During the tutorial session, several other experimenters moved around the classroom helping individual students as needed.

$$\frac{10}{b-9} = \frac{-5}{b-6} \tag{C1}$$

$$19c - 16c = 15 \tag{C2}$$

$$\frac{10}{b-9} = \frac{-5}{b-6} \tag{C3}$$

$$4 + 5(a+7) = 4 \tag{C4}$$

$$4 + 2q = 8q - 8 \tag{C5}$$

The main activity was presented as a race: the class was divided into two teams (*Red* and *Blue*) based on a random assignment of teams to identification numbers made prior to the study. One group was assigned as the GM group, the other as the paper and pencil group, and were told that they would have 15 minutes to complete as many problems (accurately) as possible, from a set of 20 total problems. These problems were similar to those in the pilot study, but were substantially easier. Equations C1 to C4 illustrate a small sample of problems from the first iteration.

The student interface was setup as shown in Figure 7 and presented problems one at a time, in an identical format. In both groups, all students were required to type in their final answer, and were given feedback about whether their answer was correct or incorrect[1]. If their answer was correct, the problem turned green, their personal score incremented by 1 on their screen, and a group score (presented on the overhead projector, and maintained by an active server connection which was also used to log result data) incremented by one. If the answer was wrong, the problem disappeared and was replaced by the next problem in the set. Students could also press a 'skip' button, with results identical to an incorrect answer. When all problems were attempted or skipped, the program began representing problems which had previously been incorrectly attempted. The only difference between the two teams was that while the GM team saw problems presented as dynamic equations, the paper and pencil team was shown visually identical but inert equations (but were naturally provided with paper and pencil). Both groups were encouraged to enter answers as soon as they knew them – that is, neither had to fully complete the problem within the provided interaction system. Once the first 15 minute competition was up, there was a planned 'rematch' with the same teams, but reversed conditions.

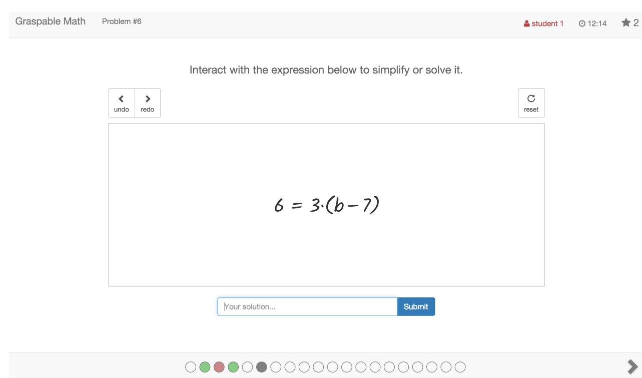

Fig. 7: The interface students used during the warm up activity, which was setup as a race between two teams. In the GM condition, the central math expression could be manipulated, while in the pen and paper condition it was inert.

This activity proved highly motivating. Most students were excited to compete, and were boisterous during the activity – there was substantial cheering, for instance, when one team took a large lead. This may have affected the extremity of the results we found. That is, as one team perceived itself to be losing, they may have worked less hard. It is worth noting, however, that in other studies students often become bored and 'drop out'. In the current study, only one student stopped participating from the group of 48, suggesting that the race format may have higher fidelity as an indicator of interface usability. Overall, we believe this to have been a highly successful method for translating interface testing into a middle-school classroom.

2) Quantitative Results: Because there were 20 problems to be solved, and a limited amount of time, the main dependent

[1] A variety of forms was accepted. For instance, for the equation $3x+2 = 11$, any of 3, $x = 3$, and $3 = x$ were accepted.

measure was the total number of problems solved correctly over the 15 minutes. A secondary measure was the proportion of submitted solutions which were correct – this is of course a practically important measure for most problem solving and testing contexts.

Figure 8 summarizes the results. For each of total solutions and accuracy rate, a 2 (grade level) x 2 (GM or PP) x 2 (order counterbalance) ANOVA was conducted. This analysis revealed that older participants achieved more correct solutions (M(8th) = 7.8; M(7th)=4.2; F(1, 46)=29, $p < 0.001$, d=1.15), and that participants using GM achieved more correct solutions (M(GM)=7.3, M(PP)=4.5, F(1,46)=46, $p < 0.001$, d=0.82). No interactions or order effects reached significance.

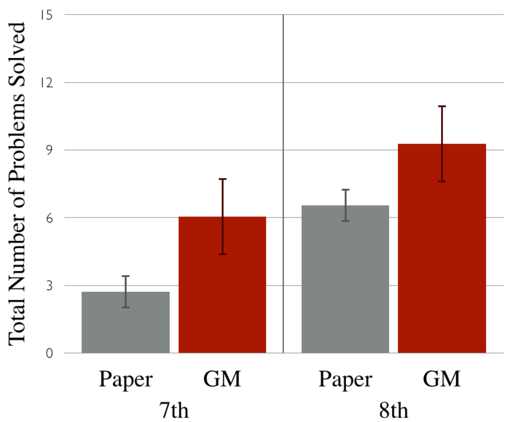

Fig. 8: Mean number of problems correctly solved in 15 minutes in the middle school usability test, collapsing across order. Error bars indicate within-bin 95% confidence interval. Because the tests were conducted within-participant, error bars do not indicate statistical significance.

Not only did students solve about 62% more of the problems within the time limit using GM, each individual answer was more likely to be correct (students could submit several answers to get a problem right). As shown in Figure 9, solutions submitted by older participants were more likely to be correct (25% vs. 45%, F(1,46)=34, $p < 0.001$, Cohen's d=0.94); so were problems solved by students using GM (45% vs. 24%, F(1,46)=38, $p < 0.001$, Cohen's d=0.91). In this case, there was also a significant 2-way interaction, such that students in the advanced classroom did much worse on paper-and-pencil problems after using GM, possibly as a result of discouragement in the competition as they perceived themselves to be losing (F(1,46)=12, p=0.001). This interpretation is compatible with comments made by the students during the competition, and with the fact that many of the error responses in this category seem to reflect random typing rather than sincere solution attempts.

It is worth noting that the error rate using the tool is still very high: fully 50% of answers entered after using GM were incorrect. A full error analysis is beyond the scope of this article, but errors seemed to come from three major sources. First, some students simply did not understand the nature of the mathematical tasks of equation solution and simplification: students often answered simplification problems

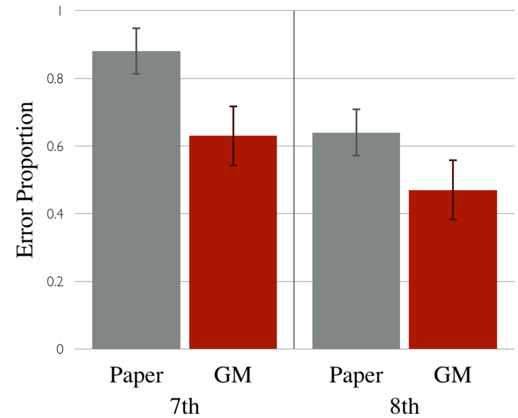

Fig. 9: Mean error rate across all problem submissions in the middle school usability test, collapsing across order. Error bars indicate within-bin 95% confidence interval. Because the tests were conducted within-participant, error bars do not indicate statistical significance.

with a solved equation, e.g., answering $x = 6$ to $8 + 2x + 4$, or with incomplete solutions, such as responding $4f = 16$ for $2f - 10 + 2f = 6$. Second, although calculation failures were less common in GM that on paper, they were not rare in GM: students frequently solved the problem only partially in the system, and made an apparent error afterward. For instance, one student turned $4 + 5(a + 7) = 4$ into $5 * a = -35$ inside GM, but then answered $a = -6$. Finally, about 2% of the errors resulted from an unfortunate decimal rounding bug in GM that we discovered during the study.

3) Qualitative Results: Students were initially extremely skeptical of the GM interface. In both classrooms, when the teams were initially assigned conditions before the first race, the GM teams audibly groaned and complained. Clearly, in their view this new interface would be more difficult, more frustrating, and, most important to them perhaps, likely to lead to slower and more error-prone solutions. In both groups, these attitudes quickly changed. Interestingly, they did not change immediately: in the 8th grade classroom (in which the results of the first race were quite close), for instance, both groups vocally estimated that the 'advantage' lay with the paper and pencil group. It was not until a few minutes into the race that people started making vocal comments about the newly perceived advantages of the GM group. By the end, essentially all students were enthusiastic about the new system.

In our revisions, we tried to increase the mathematical fluidity of the system by incorporating suggestions from our earlier pilot subjects. Reports from students suggested that these changes were largely successful. Overall, participants felt very positively about smooshing and chaining, and did not feel that the system limited their ability to take shortcuts. Most participants reported that they were able to take shortcuts as effectively in GM (over this domain) as on paper. However, some participants now reported a converse problem: the system could take several actions in a row, very quickly. Sometimes, these were so fast the user could not follow them, and did not feel that they had initiated the actions. In the extreme, this led to a lack of 'trust' that the system was behaving correctly, and a lack of confidence in their own understanding of what was happening. These observations led us to adjust how some of GM's math actions are triggered. Instead of performing an action immediately when the dragged term enters its target area, the action is only triggered if the user holds the term there for a short while. This solution still allows for convenient action chaining, while avoiding many situations of unintended action triggering.

D. Discussion

We found that working with dynamic algebra systems can have interesting implications for how we think about math and mathematical reasoning. We want to give two examples here. The first is a consequence of mentally interpreting algebraic terms as objects or using direct manipulations interfaces to interact with them. In doing this, we turn something our brain is bad at, applying arbitrary abstract rules, into something that our brain is good at, recognizing patterns, modeling simple dynamics of the world, and manipulating objects in the environment [15]. Seen from this perspective, the typical distinction between "abstract" formal notations and the "concrete" situations that they might model breaks down. Instead, mathematical notation, without relating to any external situation, is both abstract and concrete in itself.

The second example is about the way dynamic algebra systems can impact students' understanding of what algebra is actually about. In the middle school study, one of the students articulated that "It [GM] does the math for you – you don't have to think at all!". Apparently, this student considered the part that GM assisted with, the correct application of low-level rules, to be all that algebra is about. In fact, even with GM the problems were quite challenging for the students, and feedback from several of them indicated that their attention was shifted towards high-level and strategy considerations.

IV. Conclusion

Many students struggle with algebraic notation and traditional computer algebra systems are not geared towards the needs of learners. In this paper we argued for the promise of dynamic algebra notation systems that retain the strengths of paper derivations while providing additional support and structure to learners.

We presented Graspable Math (GM) as one example of such a system and discussed several design decisions that contributed to developing it into a usable, desirable system. GM uses a direct manipulation interface for algebra derivations, allows efficient chaining of actions, can trace terms through derivations, and has a workspace that lets the user construct, annotate, and connect expressions. At this point, our user studies suggest that we have built a system that many users enjoy using, at least for one class of formal derivations, and which protects them from errors while allowing a speed, fluency and user-agency comparable to paper and pencil derivations.

We believe that dynamic algebra notation systems (DANS) like Graspable Math will have a big impact on math education and e-learning tools. DANS can allow novices to explore and play with algebra expressions in a way that on paper is only possible for experts. They can supplement traditional paper as a learning tool, and shift the focus from correctly applying individual rules to high-level and strategy considerations. DANS can allow teachers to explore a record of student work, in contrast to many existing learning systems that provide teachers with the final answers only. Finally, for researchers DANS provides a new way to record, visualize, and analyze rich data about students mathematical thinking and process – something that few technology tools can do at scale.

So far, little research has been done on the effects of dynamic algebra notation systems on learners and teachers. We hope that Graspable Math can help in advancing such research. We based the Graspable Math system on web technology to make it easy to access for learners, educators, and researchers, as well as easy to integrate into existing web-based e-learning systems. We are excited about the future of dynamic algebra notation systems, and the impact they will have on how we learn, teach, reason about, and share math!

References

[1] A. Bunt, M. Terry, and E. Lank, "Friend or foe?: examining cas use in mathematics research," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2009, pp. 229–238.

[2] D. H. Landy and R. L. Goldstone, "How much of symbolic manipulation is just symbol pushing," in *Proceedings of the 31th Annual Conference of the Cognitive Science Society.* Citeseer, 2009, pp. 1318–1323.

[3] D. Landy, C. Allen, and C. Zednik, "A perceptual account of symbolic reasoning," *Frontiers in psychology*, vol. 5, 2014.

[4] P. J. Kellman, C. M. Massey, and J. Y. Son, "Perceptual learning modules in mathematics: Enhancing students pattern recognition, structure extraction, and fluency," *Topics in Cognitive Science*, vol. 2, no. 2, pp. 285–305, 2010.

[5] T. J. Cossairt and J. J. LaViola Jr, "Setpad: A sketch-based tool for exploring discrete math set problems," in *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*. Eurographics Association, 2012, pp. 47–56.

[6] R. Zeleznik, A. Bragdon, F. Adeputra, and H.-S. Ko, "Hands-on math: a page-based multi-touch and pen desktop for technical work and problem solving," in *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. ACM, 2010, pp. 17–26.

[7] A. Mendes, R. Backhouse, and J. F. Ferreira, "Structure editing of handwritten mathematics: Improving the computer support for the calculational method," in *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*. ACM, 2014, pp. 139–148.

[8] J. J. LaViola Jr and R. C. Zeleznik, "Mathpad 2: a system for the creation and exploration of mathematical sketches," in *ACM SIGGRAPH 2007 courses*. ACM, 2007, p. 46.

[9] Y. Long and V. Aleven, "Gamification of joint student/system control over problem selection in a linear equation tutor," in *Intelligent Tutoring Systems*. Springer, 2014, pp. 378–387.

[10] E. Andersen, S. Gulwani, and Z. Popovic, "A trace-based framework for analyzing and synthesizing educational progressions," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2013, pp. 773–782.

[11] E. Ottmar, D. Landy, and R. L. Goldstone, "Teaching the perceptual structure of algebraic expressions: Preliminary findings from the pushing symbols intervention," in *The Proceedings of the Thirty-Fourth Annual Conference of the Cognitive Science Society*, 2012, pp. 2156–2161.

[12] ——, "Getting from here to there!: Testing the effectiveness of an interactive mathematics intervention embedding perceptual learning," in *The Proceedings of the Thirty-Seventh Annual Conference of the Cognitive Science Society*, 2015.

[13] E. L. Hutchins, J. D. Hollan, and D. A. Norman, "Direct manipulation interfaces," *Human–Computer Interaction*, vol. 1, no. 4, pp. 311–338, 1985.

[14] R. L. Goldstone, D. H. Landy, and J. Y. Son, "The education of perception," *Topics in Cognitive Science*, vol. 2, no. 2, pp. 265–284, 2010.

[15] E. Hutchins, *Cognition in the Wild*. MIT press, 1995.