The Characteristics of Writing Environments for Mathematics: Behavioral Consequences and Implications for Software Design and Usability

Davood G. Gozli[1], Marco Pollanen[2], and Michael Reynolds[1,⋆]

[1] Dept. of Psychology, Trent University, Peterborough, ON, K9J 7B8, Canada
[2] Dept. of Mathematics, Trent University, Peterborough, ON, K9J 7B8, Canada
marcopollanen@trentu.ca or michaelchanreynolds@trentu.ca

Abstract. Effective communication and collaboration of symbolic and quantitative knowledge requires the digitization of mathematical expressions. The multi-dimensionality of mathematical notation creates a challenge for mathematical software editors. There are two different approaches for handling the multi-dimensionality of mathematical notation: either using a two-dimensional writing environment in which symbols can be placed freely (unit-based) or using an environment in which single-dimensional structural elements can be nested (structure-based). The structure-based approach constrains how users write expressions. These constraints may conflict with how mathematics is normally written. A study is reported that examines how users write mathematical expressions using two graphic based editors: one that is structure-based and one that allows the free-form manipulation of selected symbols in a diagrammatic fashion (unit-based). The results are contrasted with how users handwrite mathematics in a physical medium and implications are drawn for future software design.

1 Introduction

Mathematical expressions are a fundamental tool for representing knowledge. The successful communication of mathematical expressions is heavily dependent on the use of visual representations. Indeed, even verbal communication of mathematics relies heavily on intermediary interfaces such as pen-and-paper or chalk-and-chalkboard. In order to facilitate knowledge transfer and the real-time communication and sharing of mathematical expressions, it is therefore critical that people be able to write mathematical expressions fluently and easily. The widespread use of digital communication technologies for knowledge dissemination, discussion and collaboration, suggests that achieving efficient communication of mathematics requires that mathematical expressions be easily digitized. The purpose of the present paper is to examine how people normally handwrite

⋆ The present research was supported by a grant to MR and MP from the Social Sciences and Humanities Research Council of Canada (SSHRC). Authors are in alphabetical order.

J. Carette et al. (Eds.): Calculemus/MKM 2009, LNAI 5625, pp. 310–324, 2009.

mathematical expressions and examine how the characteristics of the writing environments affect how people write.

The digitization of mathematics is usually achieved using personal computers and relies on software programs to overcome limitations in the hardware interface. There are two major challenges for mathematics; (1) a large symbol set (the symbol problem) and (2) mathematical notation has a two-dimensional layout (the layout problem). Broadly speaking there are three different approaches to writing digital expressions; (1) use a text-based description, (2) use a digital pen, or (3) use a palette-based graphic editor. Next we briefly discuss how each of these approaches has solved the symbol problem and the layout problem.

The Symbol Problem

One solution to the symbol problem is to use a keyboard to write text-based synonyms of the graphic symbols. For example, `{\cal F} \cap \Upsilon`, is the TEX [5] representation for $\mathcal{F} \cap \varUpsilon$. The keyboard interface is quick and efficient for this form of writing. However, the mapping between the text-based and the conventional graphic-based representation is not always transparent. Therefore, efficient use of this approach to writing mathematical expressions requires learning a potentially large lexicon of terms (e.g., MathML [1], provides access to over 2,000 symbols). Another solution is to use a keyboard and mouse in combination to select symbols from graphical palettes. This method allows for a transparent mapping between the visual representation of symbols written on paper and the ones written in the digital environment. One obstacle for this type of interface is that graphical symbols compete for a limited amount of space on the display, consequently (1) only a subset of mathematical symbols may be visible at any one time and (2) the symbols may be organized in a manner that is not immediately intuitive. Consequently, it can take a long time to enter even a simple expression if the appropriate symbols cannot be located right away. A third solution to the symbol problem is to use a digital pen so that the user can enter the symbols directly into the digital writing environment. This avoids many of the problems associated with the other two interfaces because the symbols are both transparent and are not hidden from the user. At present, these interfaces are still being refined so that they can correctly recognize the full range of user-drawn symbols (e.g., $\cdot, \bullet, O, o, 0, \ ^\circ, \circ, \odot, \otimes, \oplus, \oslash, \emptyset, \phi, \ominus, \theta, \varTheta, \cdots,$), which is a formidable task in computing.

The Layout Problem

Text-only programs tend to use a writing environment that requires that expressions have a one-dimensional layout (i.e., a single string of characters). Consequently, they use nested grouping symbols (e.g., brackets & parentheses) to create sub-expressions so that a formula typically written with a two-dimensional layout can be written in a single dimension. For example,

```
\frac{\frac{a}{b}}{\frac{c}{d}+\frac{e}{f}}
```

is a TEX representation for $\frac{\frac{a}{b}}{\frac{c}{d}+\frac{e}{f}}$. The mapping between this one-dimensional and its traditional two-dimensional layout is not transparent. Thus, users are required to develop a certain level of expertise with the syntax of the language before being able to use the technology properly. In addition, given that the two-dimensional layout often conveys metaphorical properties of an expression, many of these will be lost in a one-dimensional representation [6].

A second solution to the layout problem is provided by digital pen-based technologies for writing mathematical expressions, such as FFES [17], which allow users to draw symbols anywhere on a virtual page. These interfaces have the promise of writing as easily as pen-and-paper with the additional benefit of having the software identify the expression. Great strides have been made in developing this interface, however, the potential for a robust interface for handwritten mathematics has yet to be achieved.

Two solutions to the layout problem have been developed for palette-based editors. The visual display of an expression in these editors has a two-dimensional layout. However, for the majority of palette-based editors the writing environment for the expression consists of nested one-dimensional structures (e.g., Microsoft Equation Editor and BrEdiMa [9]). For example, a fraction is typically created by selecting a fraction structure from the palette. The fraction structure usually inserts a fraction bar into the expression that is bound to empty one-dimensional writing slots above and below it (see [11]). Each slot may then be populated with their own nested sub-expressions. These slots are sub-divisions of the main writing space and are often indicated with outline boxes or background shading. While the graphical presentation of the expression may be two-dimensional for the reader, for the writer this approach can be thought of as a simple extension of the text-only method (i.e., indirect access to the two-dimensional layout) with grouping symbols replaced by slots. *This type of writing environment affords a structure-based writing style* because it constrains the order in which symbols are added to an expression by giving precedence to symbols that affect the expression's layout. Creating a correct visual representation of an expression, therefore, often requires understanding the deep layout structure of the expression and which symbols parse the physical layout of the expression, before writing.

A second solution that exists in palette-based editors is to allow users to "draw" their expressions by placing symbols on a virtual canvas with direct access to a two-dimensional space (i.e., XPRESS [12]). Once the expression is drawn a spatial analysis algorithm, similar to those from pen-based systems, is applied to identify the expression. As the symbols are chosen from palettes and "placed" by the user on a virtual canvas, there is little doubt about the identity of the symbols and their intended locations. This greatly reduces the complexity of expression identification as compared to handwritten mathematics. It also reduces the complexity of editing expressions in that in a two-dimensional space, items are directly accessible and users are not required to first choose among spatial units and then particular items. The two-dimensional writing environment allows users to select symbols from palettes in any order and place them anywhere they wish. *This type of writing environment affords a unit-based writing style* because it does not place any constraints on the order in which symbols are written.

In summary, at least three writing environments have been developed as solutions to the layout problem (1) a one-dimensional layout with multiple embedded substructures, (2) a two-dimensional layout, and (3) a two-dimensional layout constructed from nesting one-dimensional structures.

The Present Study

Although many studies have examined different solutions to writing digital expressions, it has not been possible to disentangle how writing is uniquely affected by solutions to the symbol and the layout problems. The recent development of Xpress, a palette-based editor with a two-dimensional writing environment, provides a unique opportunity to examine how different solutions to the layout problem affect writing behaviour. The purpose of the present study, therefore, is to examine how the characteristics of a writing environment affect how people write mathematical expressions. Here we focus on two different writing environments that are palette-based.

Although writing environments have their own set of rules governing the types of actions that are permitted and how space is allocated, it is unclear whether this will result in *actual* behavioural differences in how mathematical expressions are written. Here we follow the suggestion that user behaviour can only be understood when (1) the types of behaviours permitted (i.e., affordance), actual use (i.e., effectivity), and goals, motives and perceptions (i.e., intentionality) are considered simultaneously [3,4]. To understand how the characteristics of a writing environment affect how people write expressions, people were observed as they wrote mathematical expressions by hand and using two different graphics based software environments. A handwriting condition was included to assess how people would write the expressions under natural conditions. The two software environments (BrEdiMa and Xpress) were selected because they use similar interface technologies (i.e., keyboard and mouse) and representations (graphic symbols) to enter mathematical expressions, and therefore only the rules governing how symbols are arranged in the environments are qualitatively different. Novice users were examined to control for expertise. If the editors require that users change their writing style, then the use of novices will allow us to document some of the challenges that they encounter.

2 Method

2.1 Subjects

Seven members of the Cognitive Ethology Lab at Trent University participated in the present study, four of which were undergraduate students, two were graduate students, and one was a faculty member. One subject was left-handed. The subjects were all familiar with simple mathematical and logical notation, although none had previous experience using either of the software editors.

2.2 Stimuli

Eight different mathematical expressions were used (see Table 1). The stimuli were split into two sets, each set was typeset and printed a separate sheet of paper. The first set (Expressions 1–4) had fewer symbols (mean=7.5) than the second set (Expressions 5–8; mean=25.5).

Table 1. The two expression sets used in the study. Expression number is indicated in parenthesis.

Expression Set 1	Expression Set 2
(1) $\overline{A \wedge B \vee C}$	(5) $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
(2) $\overline{A \wedge \overline{B \vee \overline{C}}}$	(6) $r_{xy} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y}$
(3) $\frac{\sqrt[3]{x^2+1}}{2x}$	(7) $r_{xy} = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}}$
(4) $\sum_{i=1}^{100} i^2$	(8) $\lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$

2.3 Apparatus

In the handwriting sessions, mathematical expressions were written using dry-erase markers on a 36” x 48” whiteboard affixed to the wall of a small conference room. Although the whiteboard environment is different than paper in a number of aspects (e.g., orientation of the writing surface, thickness of the writing instrument, and physical size of the written expression), the principles of the writing environments are assumed to be the same for the whiteboard and paper. The writing sessions were video recorded using a Canon HG10 video camera. The camera was operated by the experimenter and hand-held to allow for adequate observation of hand movements and written symbols.

In the software writing sessions, mathematical expressions were written in a small office using BrEdiMa and Xpress. Comparing BrEdiMa and Xpress allowed us to control many input and representation features and to isolate the difference between one- and two-dimensional writing environments. Both are standalone browser-based editors, and both have an AJAX-based front-end that allows users to create their expression and then submit their expression to a server which returns a LaTeX preview. None of the subjects had used either interface before. Although the palettes in Xpress contain more symbols than those in BrEdiMa (138 to 50), both are minimal editors relative to commercial alternatives.

Data was collected using an Acer personal computer with a core-2-quad processor, a 19" LG Flatron LCD screen and running Windows XP operating system. The editors were run inside the Mozilla Firefox Web browser (version 2.0.0.16). The writing sessions were recorded using in video format using SnagIt 9.0 by TechSmith (http://www.techsmith.com), which recorded all occurrences on the computer screen during a writing trial.

2.4 Procedure

Data collection was spread over six separate sessions (2 expression sets $\times$ 3 writing environments) that lasted approximately 20 minutes each. Thus each mathematical expression was written three times (once for each interface) and each interface was used on two separate occasions (once for each expression set). The conditions were always completed in the same order. The short expression set was used in sessions 1–3 and the long-expression set was used in sessions 4–6. Furthermore, the order of the writing environment was the same for each set, first with the whiteboard, then BrEdiMa, and finally Xpress. On average, there was a 24-hour interval between each session, for each subject. The order of the sessions was chosen to control transfer between writing platforms. For instance, having the subjects use BrEdiMa before Xpress enabled us to observe the way subjects prefer to write in the latter environment after being exposed to both writing methods.

At the beginning of each session, subjects were given a sheet containing the expressions to be written. Subjects were able to refer to the expressions before and during the writing process. Subjects were instructed to write the expressions one at a time and in the order on the sheet. They were encouraged to go quickly, but not to sacrifice accuracy for speed.

In the handwriting condition, subjects were instructed to signal the experimenter when they were starting an expression and when they were finished writing an expression. The whiteboard was erased between expressions, so that only one expression appeared on the whiteboard at any one time.

In the software editor conditions, the editors were loaded via the Internet before subjects arrived in the lab. Subjects were not given any instruction concerning how to use the interfaces, nor regarding the specific use of keyboard or mouse. However, they were asked to re-load the software using the browsers refresh button, after completing each expression, so that only one expression appears on the screen at any time.

3 Results

Data analysis began by coding over 4,800 discrete behavioural events in the video recordings. An event was defined as any action that had a direct effect on the mathematical symbols represented in the writing environment (e.g., adding a symbol). The timing of an event was linked to when changes occurred in the writing environment. An event did not need to be a correct step towards a successful

Table 2. Overall writing time (in seconds), number of events and time per event for each writing platform and as a function of each type of action event

	Whiteboard	BrEdiMa	Xpress
Overall			
Total Time	168.9	1007.1	951.6
Number of Events	138.4	218.3	197.1
Time per event	1.2	4.5	4.8
Adding Units			
Total Time	164.9	852.3	684.0
Number of Events	135.9	196.0	152.3
Time per event	1.2	4.3	4.5
Deleting Units			
Total Time	1.4	154.9	37.4
Number of Events	0.7	22.3	10.7
Time per event	2.0	6.9	3.5
Modifiying Units			
Total Time	2.6	NA	230.1
Number of Events	1.9	NA	34.1
Time per event	1.4	NA	6.7

completion of a desired formula. Three broad classes of events were identified; Addition events consisted of all events directly required to add elements to the display; Deletion events included any action that was directly implicated in the removal of a unit; and Modification events included any action that directly changed either the spatial location or physical appearance of an element in the display. The data were analyzed using a repeated measures ANOVA with writing platform (Whiteboard, BrEdiMa, and Xpress) as the repeated factor. Unless otherwise specified the degrees of freedom for all tests are 2 (treatment) and 12 (error) and significant findings are reliable at the .05 level.

As can be seen in Table 2, the writing environments differed dramatically in how long it took users, on average, to write the expressions ($F = 36.4$, $MSE = 42285$). Mathematical expressions were written fastest in the handwriting condition, followed by BrEdiMa and Xpress, which were not reliably different ($t(6) < 1$). To better understand why users took longer to write the expressions using BrEdiMa and Xpress we further examined the number of events that occurred during the writing process and how long users spent per event.

We assumed that changes in the number of events made while writing would indicate a change in the writing process. Consequently, we hypothesized that if the number of events changed across writing environments, then this would indicate that the properties of the different writing environments affected how people write the expressions. Consistent with users changing how they write expressions when using the software interfaces, they required substantially more events compared to the handwriting condition, $F = 21.4$, $MSE = 560$. Although BrEdiMa had 21 more events on average per subject than did XPRESS, the difference was not reliable ($t(6) = 1.5$, $p > .10$).

We also hypothesized that the average time per event would provide insight into how easily users were able to interact with the environment. Unsurprisingly, the average duration of an event was also substantially longer for the software editors compared to the handwriting condition, $F = 64.2$, $MSE = .429$. Given that BrEdiMa and XPRESS are both palette-based editors we anticipated that they would not differ on this dimension. Consistent with the difficulty of navigating the palettes being similar, the software editors did not differ in the average duration of an event, ($t(6) = 1.1$, $p > .10$).

Types of Events

In order to better understand how users were writing the expressions we examined performance as a function of the types of events that people engaged in. Three types of events were examined Additions, Deletions and Modifications.

Addition and Deletion Events

As can be seen in Table 2, users spent substantially more time adding and deleting symbols when using the software editors than they did when using the whiteboard ($F= 28.7$, $MSE = 31333$ and $F=14.0$, $MSE = 3223$, respectively). More time was spent adding and deleting symbols with the software editors because (1) the users made more addition and deletion events with the software editors, ($F = 12.4$, $MSE = 545$ and $F = 25.5$, $MSE = 31.9$, respectively), and (2) it took users more time to execute addition and deletion events with the software editors ($F = 50.0$, $MSE = .467$, and $F = 4.9$, $MSE = 22.6$, respectively).

In order to examine how the characteristics of the writing environments affected performance independent of interface type (pen vs. palette), we compared performance for BrEdiMa and XPRESS. We expected that XPRESS would require fewer addition events because the one-dimensional canvas used in BrEdiMa often requires the user to add new spatial locations for those elements that do not belong to the same structural domain (e.g., a suprascript location). Consistent with this hypothesis, 44 more addition events per subject were made with BrEdiMa than XPRESS ($t(6) = 2.8$, $p < .05$).

Furthermore, there were 12 more deletion events per subject when using BrEdiMa compared to XPRESS, consistent with giving precedence to structure symbols (for creating new one-dimensional writing canvases) increasing the difficulty of writing expressions ($t(6) = 3.3$, $p < .05$). Interestingly, despite taking twice as long to make a deletion event in BrEdiMa compared to XPRESS, the difference was not reliable ($t < 1$). The reason was a large amount of variability in duration of the deletion events in BrEdiMa. As it turns out, deleting or changing parts of an expression in a structure-based environment requires selecting the appropriate space. This can cause confusion since being in one space makes other spaces inaccessible for editing. In response to this inaccessibility, users tended to clear the writing environment (by refreshing the browser) and start over instead of removing an unwanted part of an expression. There were 23 re-starting events in BrEdiMa (more than 3 per subject), by comparison there were none in either XPRESS or handwriting conditions.

Writing Order

The nature of the writing environment can also affect a users writing style. As noted above, a two-dimensional writing space affords a unit-based writing style because symbols can be written in any order and placed at any location. In contrast, constructing a two-dimensional spatial layout using nested one-dimensional canvases affords a structure-based writing style because precedence must be given to symbols that affect the spatial layout of the expression. These predictions were assessed by examining how much variability in the order that symbols were added to the expression (writing order) was explained by the unit- and structure- based writing styles. The initial writing order was used as a measure of how people *attempted* to write the expression. The final writing order was used as a measure of how people ended up writing the expression. The variance explained by each writing style was calculated independently for each subject and each formula.

Our implementation of the unit-based writing style presumed writing order would be left-to-right and top-to-bottom (as opposed to random). Our rationale was that (1) equations are typically read left-to-right, top-to-bottom irrespective of the direction of a cultures text-based writing (e.g., Persian), and (2) this structure is argued to be linked to peoples understanding of the mathematical relationships (see [6]). Similarly, our implementation of the structure-based writing style presumed that precedence would be given to only those symbols that were required to add new one-dimensional writing slots (thus single line operators such as $*$, $+$, or $\div$ were not seen as having special priority). It was assumed that within the one-dimensional structures, a unit-based approach would be employed.

Handwriting. As can be seen in Table 3, the unit-based writing style best captured overall writing order, $F(1, 6) = 26.1$, $MSE = .211$. Indeed, only one person, a computer science major, wrote using the structure-based style. Given that the whiteboard interface very closely approximates writing with a pen and paper we expected that people would not change their writing style while writing an expression. Consistent with this prediction, the unit-based writing style captured both peoples initial- and final- attempts to write an expression equally well ($F(1, 6) = 3.7$, $p > .10$, $MSE = .004$). In order to assess whether people adjusted their writing style, but did so only once while writing the first expression, we examined performance for Expressions 1 and 2 more closely. The unit-based method accounted for 86% of the variability in Expression 1 and 87% in Expression 2, whereas the structure-based method explained only 16% of the variability in Expression 1 and 8% of the variability in Expression 2. Furthermore, there was difference between initial and final writing order ($F<1$), consistent with people not needing to adjust their writing style with this writing environment because it is similar to paper.

Table 3. Amount of variability (R^2) in initial- and final- symbol placement order as a function of writing style (unit-based vs. structure-based) and writing environment

Interface	Attempt	Unit-based	Structure-based
Whiteboard	Initial	.84	.53
	Final	.83	.51
BrEdiMa	Initial	.55	.78
	Final	.41	.92
XPRESS	Initial	.78	.57
	Final	.77	.52

An informal analysis of peoples writing behaviour revealed that deviations from the writing order predicted by the unit-based style primarily arose from violations of our left-to-right and top-to-bottom assumption. For instance, in Expression 3 people would often write the index of the radical after writing the radicand. Another violation is captured with Expression 4 (which was unique in that it was the only expression for which the unit and structure-based styles predicted the same writing order). Despite both writing styles making the same predictions, they only explained 74% of the variability in writing order. The reason: people added the "$\sum$" first and then were essentially random as to whether they would add the initial condition or the upper bound portions of the expression next.

BrEdiMa. As expected, writing order was best captured by the structure-based writing style, F(1, 6) = 683.4, MSE = .011. Unlike the handwriting condition, writing style tended to change as people wrote each expression. The unit-based writing style best captured peoples initial attempt to write an expression, whereas the final attempt was best captured by the structure-based style, F(1, 6) = 66.3, MSE = .016. This is consistent with people changing their writing style to give precedence to structure symbols that create the two-dimensional layout.

In order to better examine how people adjusted their writing style, we examined performance for Expressions 1 and 2 more closely. The unit-based writing style best captured the initial attempt at writing Expression 1, explaining 84% of the variability compared to 18% for the structure-based style. The final attempt at writing the expression was best explained by the structure-based writing style (100%) compared to the unit-based style (2%). This suggests that people quickly and efficiently adjusted to the demands of writing a two-dimensional expression using nested one-dimensional slots. This change in writing style persisted when users wrote Expression 2, which is of the same general form as Expression 1. This time, the unit-based style explained less than 1% of the variance in performance whereas the structure-based account explained 99% and the structure based account captured both the initial- and final- attempts. For each of the remaining expressions, however, (except for Expression 4, see below) there was a change between the initial and final writing order. In each case there was a reduction in the explanatory power of the unit-based writing style over time, with the structure-based method increasing in explanatory power. This suggests that subjects initial approach is to employ a unit-based writing style and adapt their approach as the situation warrants.

Deviations from the writing order predicted by the structure-based writing style primarily arose from two sources (1) when individual symbols need to be corrected (see below) and (2) when a symbol added more than one writing dimension and placed the cursor at an unpredicted location. One example of this is captured with Expression 3 in which the cursor was placed inside the radicand instead of at the index location.

Xpress. Similar to the handwriting condition, writing order was best captured by the unit-based writing style, $F(1, 6) = 8.7$, $MSE = .297$. It is important to highlight that this was true despite carry over effects from having used the structure-based editor in the previous session. The amount of variability explained by both writing styles decreased from the initial attempt to the final attempt at writing an expression, $F(1, 6) = 17.4$, $MSE = .024$. This reduction in the explanatory power of both approaches was related to people modifying the appearance of symbols (see below).

Once again we examined performance for Expressions 1 and 2 separately. The initial writing order for Expression 1 was best explained by the unit-based writing style (70%) compared to the structure-based style (30%). The unit-based writing style captured even more variance in the final writing order (82%) compared to a decrease in the structure-based style (22%) suggesting that users were once again changing their writing style. This is consistent with some carry over from writing with a structure-based in BrEdiMa, but that people ultimately preferred the unit-based writing style. This change appears to have stabilized as early as Expression 2, were the unit-based writing style accounted for 82% of the variance in symbol placement order compared to 12% for the structure-based style. There was no difference between the initial and final writing order.

Deviations from the unit- and structure- based writing styles predominantly arose from violations of our assumption that people would write left-to-right and top-to-bottom and were very similar to the violations that occurred in the handwriting condition. Another source of error, above and beyond those observed with in the handwriting condition, concerned the modification of symbols. Sometimes people would replace a symbol if it was not an appropriate size.

Modification Events

Modifications accounted for approximately 1% of all events in the handwriting condition and 17% of all events for Xpress. Modifications were not observed for BrEdiMa because the overall structure and the spatial relations among symbols are determined automatically by the structure-units that specify the spatial layout. Compared to the handwriting condition, substantially more time was spent modifying symbols in Xpress, ($F(1, 6) = 41.9$, $MSE = .362$). This increase in time was a consequence both of more modification events, ($F(1, 6) = 77.7$, $MSE = .004$), and more time being spent per modification event, ($F(1, 6) = 63.6$, $MSE = 1.17$). Modification events within the handwriting condition typically consisted of extending the length of horizontal lines. In contrast, almost all types of symbols were resized in XPRESS. In XPRESS it was possible for users to change the structural position of a symbol by dragging it to a new location. Such changes in spatial location would require a deletion and an addition event when using either the whiteboard or BrEdiMa. If the majority of modification events were of this type, then this could have important implications for how we understand the consequences of having to change writing styles. The data were therefore reanalyzed to examine how many events involved changes in layout that were structural in nature and that could be conceptualized as a simple deletion-addition event when using the whiteboard, BrEdiMa and XPRESS (due a movement in the structural position of a symbol as opposed to subtle changes in relative spacing). This analysis revealed that no such events occurred with the whiteboard, 1.7 events with BrEdiMa and 2 events with XPRESS. Together, these data suggest that no change is required in how the addition, deletion, and modification data are understood.

4 General Discussion

The purpose of the present study was to examine how differences in the characteristics of writing environments affect how people write mathematical expressions. Handwriting and writing with the two-dimensional software environment were largely characterized by a unit-based writing style in which individual symbols were added in a left-to-right, top-to-bottom fashion. In contrast, writing with the one-dimensional software platform was characterized by a structure-based writing style in which precedence was given to symbols that created additional one-dimensional writing spaces. Thus, the indirect access to the two-dimensional writing space led users to change how they write. Although users were able to adjust to the demands of the structure-based writing style, there is evidence that it was less than intuitive. First, users seemed to adjust their writing style on an as-needed basis, after encountering problems. This suggests that they are able to remember specific instances were they have had to adjust, but have difficulty generalizing this knowledge to new contexts. Second, users found the environment difficult to navigate; this was most evident in the number of times symbols were deleted and the number of times users elected to rewrite an expression from scratch rather than fix an error. One concern is that the increased cognitive load that arises from having to operate in an unfamiliar writing environment may result in more dramatic performance deficits in time-pressured situations [16]. For instance it is well known that reading performance is dramatically impaired when having to perform a second task, even if it requires independent sensory and effector systems (e.g., [10,13]).

Although the two-dimensional software environment was more intuitive than the one-dimensional editor, users spent approximately the same amount of time using both editors. Users of the two-dimensional writing system spent a substantial amount of time modifying the display. The fact that 17% of the time using XPRESS was spent adjusting the cosmetic features of the users input, despite the software correctly recognizing the correct equations and reformatting the final output through LaTeX, suggests that sizing is an important issue that needs to be addressed in two-dimensional environments. The sizing issue is largely avoided when users handwrite an expression because they tend to draw symbols at an appropriate size. The size of a symbol is typically not an issue for most structure-based editors because sizing is accomplished automatically through the nested structure of the environment. One solution, therefore, to this problem in two-dimensional software environments might be to analyze structure on-the-fly to provide automated assistance with symbol sizing and association.

At present it is unclear how and whether using a pen-based digital interface will result in modification events. Given that handwriting seldom results in modifications (we did not observe any here), we anticipate that modifications would have to arise as a consequence of the handwriting recognition process. Failures recognizing written symbols will require users to clean-up or resize what they have written. Similarly, an error in layout analysis might require a user to manually resize, delete, and/or move a symbol in a written expression.

Although we discussed two violations of the left-to-right, top-to- bottom writing order, more occurred. Documenting violations of the "normal" reading and writing order may help develop a more intuitive structural interface and provide insight into the cognitive factors that influence mathematical writing. In some instances, violations occurred because the standard form of the expression violated the standard writing order (yet some people still apply the left-to-right and top-to-bottom order), as was true of the summation operator. In other instances, it is less clear why violations occurred. For instance, users may have waited to write the index in the third-root component of Expression 4 because (1) users understanding of the expression is incomplete, (2) there is forward momentum in the writing process, (3) users conceptualize the index as analogous to an accent, or (4) users are most familiar with writing the square root, which does not require an index. Presently there is insufficient evidence to discriminate among the many alternatives. One important objective for future research, therefore, is to more thoroughly document when violations of the left-to-right, top-to- bottom assumption occur.

Two additional lines of inquiry that are relevant for software development concern (1) writing expressions from memory and (2) how writing environments affect learning. For instance, in the handwriting condition people wrote brackets in the order they appeared. However, people may wait until all of the necessary symbols are written before using brackets to nest the symbols when writing from memory. With respect to learning, one-dimensional editors may improve fluency in mathematics because they require users to understand the deep structure of an expression. Consistent with this possibility, the nature of a writing technology has been shown to affect how people think about the material they are writing [2]. Interestingly, it may be possible to improve a user's understanding by implementing visual coding of structure given that the abstract meaning of an expression is cognitively related to the details of its actual representation [7,8]. Given that the size of symbols and their relative spacing are often used in mathematics to perceptually organize an expression it may be possible to examine the relative importance of these visual cues by examining how users make cosmetic changes to their written expressions using XPRESS (e.g., $4+4\ \times\ 2+2$ vs. $4\ +\ 4\times 2\ +\ 2$).

In the present study we examined how the writing environment affects writing behavior, and did not focus on the usability of particular interfaces. However, for future work, this does raise the issue that there are no standard methodologies and benchmarks for the scientific comparison of mathematical input interfaces. In comparison, the usability of text-input interfaces has been well studied. For example, there are widely used methodologies like the Roberts and Moran Methodology [14] that examine the usability of text editors in terms of time, error and learning. Our study does raise the issue about whether a methodology for comparing mathematical input interfaces should also consider if a mathematical interface forces changes in a users writing behavior, potentially increasing their cognitive load, irrespective of differences in performance measures such as time and accuracy.

5 Conclusions

Traditionally interfaces for mathematical expression entry were found mainly in document creation environments and computer algebra systems. The explosion of Web-based technologies has created a demand for new applications, such as online collaboration and assessment tools, many of which can be considered real-time applications. At the same time there have been several recent attempts (e.g., XPRESS, pen-computing) to develop two-dimensional mathematical writing environments. In this paper we have shown that two-dimensional environments allow users to write mathematical expressions in a more intuitive way than one-dimensional environments. Therefore, continued research into two-dimensional interfaces may have important implications for the development of future mathematical interfaces, especially real-time ones, where the main goal is to communicate quickly and effectively.

References

1. Carlisle, D., Ion, P., Miner, R., Poppelier, N., et al.: Mathematical Markup Language (MathML) Version 2.0. W3C Recommendation (February 21, 2001) `http://www.w3.org/TR/2001/REC-MathML2-20010221/`
2. Edwards, L.D.: Embodying mathematics and science: Microworlds as representations. Journal of mathematical Behavior 17, 53–78 (1998)
3. Gibson, J.: The Ecological Approach to Visual Perception. Lawrence Erblbaum Associates, Inc., Hillsdale (1986)

4. Kadar, E.E., Effken, J.: From discrete actors to goal-directed actions: Toward a process-based methodology for psychology. Philosophical Psychology 18, 353–382 (2005)
5. Knuth, D.E.: The TeX book. Addison-Wesley, Reading (1984)
6. Lakoff, G., Núñez, R.: Where Mathematics Comes From: How the Embodied Mind Brings Mathematics into Being. Basic Books, New York (2000)
7. Landy, D., Goldstone, R.L.: How abstract is symbolic thought? Journal of Experimental Psychology 33, 720–733 (2007)
8. Landy, D., Goldstone, R.L.: Formal notations are diagrams: Evidence from a production task. Memory and Cognition 35(8), 2033–2040 (2007)
9. Nakano, Y., Murao, H.: BrEdiMa: Yet Another Web-browser Tool for Editing Mathematical Expressions. In: Proceedings of MathUI 2006 (2006) (online)
10. O'Malley, S., Reynolds, M.G., Stolz, J.A., Besner, D.: Reading aloud is not automatic: Lexical and sub-lexical spelling to sound translation use central attention. Journal of Experimental Psychology Learning Memory and Cognition 34, 422–429 (2008)
11. Padovani, L., Solmi, R.: An Investigation on the Dynamics of Direct-Manipulation Editors for Mathematics. In: Asperti, A., Bancerek, G., Trybulec, A. (eds.) MKM 2004. LNCS, vol. 3119, pp. 302–316. Springer, Heidelberg (2004)
12. Pollanen, M., Wisniewski, T., Yu, X.: XPRESS: A Novice Interface for the Real-Time Communication of Mathematical Expressions. In: Proceedings of MathUI 2007 (2007) (online)
13. Reynolds, M., Besner, D.: Reading aloud is not automatic: Phonological recoding and lexical activation use central processing capacity. Journal of Experimental Psychology: Human Perception and Performance 32, 799–810 (2006)
14. Roberts, T.L., Moran, T.P.: The evaluation of text editors: methodology and empirical results. Communications of the ACM 26, 265–283 (1983)
15. Smithies, S., Novins, K., Arvo, J.: Handwriting-Based Equation Editor. In: Proceedings of Graphics Interface 1999, pp. 84–91 (1999)
16. Vicente, K.: The Human Factor: Revolutionizing the Way People Live with Technology. Knopf, Toronto (2003)
17. Zanibbi, R., Blostein, D., Cordy, J.R.: Directions in Recognizing Tabular Structures of Handwritten Mathematics Notation. In: Proc. Fourth Int'l IAPR Workshop on Graphics Recognition, pp. 493–499 (2001)