Abstract. Computer software interfaces for mathematics collaboration and problem solving rely, as all interfaces do, on user identification and recognition of symbols (via icons and other contextual widgets). In this paper we examine the results of a short study which examined users interacting with mathematics software (Mathematics Classroom Communicator, MC2) designed for education, real-time communication and collaboration. Videos were recorded of 14 users working through seven comprehensive problems in the MC2 interface. Extensive second-by-second coding was completed of the user’s actions and status throughout their work, and a set of transition matrices were tabulated, estimating transition probabilities between symbols, operators and other aspects of mathematical expressions. We discuss the results of these matrices, and their implications in the translation of abstract mathematical concepts into software interfaces, and further conclude with a brief discussion of suggestions for mathematical software interface design. This study also has applications in mathematical software usability and accessibility.

Keywords: Mathematical notation · Eye-tracking · Mathematical software interfaces · Transitions

1 Introduction

Computer software applications which support the transcription of mathematics make use of a number of models to represent mathematical structure. For most experts, the 1-dimensional model used in \LaTeX is preferred once time is taken to learn the syntax due to the power and expressiveness of the system. In \LaTeX the transcription proceeds as a single string of characters, transformed into possibly complex two-dimensional mathematical structures upon compilation. Due to the lack of immediate connection between this 1-dimensional \LaTeX code and the compiled output, and the accompanying learning curve, this approach can present significant challenges for novice users. Due to the extreme market penetration of Microsoft Word, many novice users simply do any required electronic
mathematical transcription using the WYSIWYG (what you see is what you get) Microsoft Word Equation Editor or similar products. These products tend to be somewhat intuitive in their use, with extensive menu and icon systems that allow for eventual (if not efficient) implementation of marginally complex mathematical expressions. Further, as WYSIWYG editors provide users with the ability to directly edit the mathematical output within the user interface without a need to recompile the document, their use by novice level users has been shown to result in fewer output errors [4]. While \LaTeX provides users with many positive features not available in WYSIWYG editors there are a large number of individuals who rely solely on WYSIWYG editors (and often are not even aware of the existence of \LaTeX). Therefore, it is important that users are able to use these editors with the highest level of efficiency and effectiveness. To improve these editors we must have a thorough understanding of how individuals interact with their respective environments.

Within the subset of WYSIWYG editors, there are different representative models for mathematical structure utilized, and it remains unclear which model provides an overall more positive user experience in their respective environments [4–6]. It is likely the case that there is not one model which is superior to others for all elements of mathematical notation, but rather that each model handles some mathematical structures better than others. To examine this, the following study was designed to compare the execution of insertion and editing of different types of mathematical structures by novice users through examination of the probabilities of symbol entry order.

\textit{MC}^2, an application created by Pollanen and Kang (co-authors on this paper), with colleagues Cater, Chen & Lee, allows users to construct expressions in a free-form manner through a 2-dimensional drag and drop interface in which symbols are inserted onto a 'blank canvas' workspace from a series of menus and can subsequently be moved and resized using the mouse. The technical details of \textit{MC}^2 specifications can be found in Pollanen \textit{et al.} [7]. As a typical default choice due to ubiquitous availability, Microsoft Equation Editor was used as a comparator. Equation Editor is a 2-dimensional structure-based editor, in which users generally insert the mathematical structures first (with imposed dimensionality, e.g., powers or subscripts), and then users interact with the input options of the structures to insert symbols in the form of 1-dimensional strings. Though users interact directly with the 2-dimensional mathematical structures in both programs, they must do so in different ways [1]. Through this study we aim to show that the model chosen within these typesetting applications changes the order in which users enter symbols into their workspace resulting in a change in their workflow, with some implications for future design.

2 Methods

The study consisted of 14 first-year undergraduate student participants (10 female and 4 male), who are not from quantitative fields, and none of whom had any previous experience using any mathematical equation software. They
were instructed to transcribe a series of four different mathematical expressions using the two digital typesetting applications (MC\textsuperscript{2} and Equation Editor). Each expression was first transcribed with MC\textsuperscript{2} and subsequently with Microsoft Equation Editor (in Word). This resulted in 8 experimental trials per participant. All participants completed the experiment simultaneously and communication with the researcher occurred through a chat screen contained within the user interface of MC\textsuperscript{2}. All participants completed the 8 trials in identical order. The expressions provided to the participants are displayed in Table 1.

Table 1. Expressions provided to participants for transcription using the computational environment.

|   | Expression                                                                 |
|---|---------------------------------------------------------------------------|
| 1 | $s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x - \bar{x})^2}$                |
| 2 | $\int \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx$                       |
| 3 | $\int_{0}^{\infty} e^{-y} dy$                                            |
| 4 | $\sqrt{\frac{\sqrt{x^2 + 2x + 2} x}{x^2 + 1}}$                          |

The video recordings (screen captures) of the 14 participants’ actions were analyzed by a research assistant, who carefully watched the videos and assigned a time-stamped code to each significant, discrete action performed by the participant while working towards completion of the transcription task. Coded actions consisted of: insertion of a symbol; deletion of a symbol; resizing of a symbol; and movement of a symbol. Each symbol in every expression was assigned a unique reference within the given expression. The assigned codes contained information indicating both the action and the unique symbol upon which the action was performed. For example, the action of inserting the symbol assigned the number 4 would have been assigned the code, “i-4”.

From this analysis, we were then able to analyze the order in which individuals inserted the mathematical structures relative to other symbols in the expression through Markov Chain transition matrices. Markov Chains were applied in a similar fashion by Jansen, Marriott, and Yelland to model the relative order in which individuals read symbols in algebraic expressions [2]. As this proved to be an effective analysis, we applied this approach to model the relative order of symbol insertion when digitally transcribing mathematical expression.

The resulting transition matrices contain the conditional probabilities that a symbol in a particular part of the expression will be inserted, given that a symbol from another part of the expression was the previously inserted symbol. Therefore, in a transition matrix $T$, the entry in position $T_{i,j}$ represents the probability that the symbol from the part of the expression indicated in column $j$ is inserted, given that the symbol from part of the expression in row $i$ was the symbol previously inserted. We initially created these transition matrices to represent the probability of transition between all individual symbols within
the four expressions. However, the experimental expressions contained numerous symbols (>20), and we therefore decided it was more clear and concise to create transition matrices which tabulated transition probabilities relative to the mathematical structure found in these expressions with a unique transition matrix. We then analyzed the transitions between structural components of the expression. To create these transition matrices, we first recorded all insertion events which occurred in each experiment video recording. We then categorized the transitions between insertions, increasing the count in \( T_{i,j} \) by one when the symbol in column \( j \) was inserted following insertion of the symbol in row \( i \). For these matrices we chose to ignore transitions occurring within the same region of an expression. Therefore, all counts of \( T_{i,i} \) were 0. We then normalized each row, so that the sum of all entries in each respective row was 1 or 0 in the case where the symbol represented by that row was the last symbol inserted in all trials. The result was a transition probability matrix for each experimental trial for a given structure, for the 14 participants in the experiment. The transition matrices from the applicable individual trials within each participant were then normalized, summed, and re-normalized, resulting in our final transition probability matrices for each of the mathematical structures found in the experimental stimuli.

The transition probability matrices for the four types of mathematical structures included in the experimental stimuli are shown in the Results below (Tables 2, 3, 4 and 5), displaying transition probabilities related to all square roots, fractions, definite integrals and summations present in the four expressions used in the study. Though the exponent structure was featured in these expressions, we did not choose to model their insertion order due to significant inconsistencies in the manner of their insertion when using MC\(^2\) and Microsoft Word Equation Editor.

3 Results

3.1 Markov Chain Transition Probability Matrices for Square Root Structures

The matrices in Table 2 highlight some key differences observed in how users inserted the square root structure into their workspace. Notably, users were about 16% more likely to insert a symbol inside of the square root immediately following insertion of the square root structure when using Equation Editor than when using MC\(^2\). Users were also about 23% more likely to insert the square root structure immediately following insertion of a symbol inside of the square root when using MC\(^2\) than they were when using Equation Editor.

From these transition probability matrices, we can compute the probability of the order of entry for an expression containing a radical when using both interfaces. One logical order of entry, if we assume the user starts from outside of the radical, for such an expression is Outside → Radical → Radicand. The probability of this symbol entry order after 2 transitions when using MC\(^2\) was approximately 0.56 versus 0.70 when using Equation Editor. On the other hand,
an alternative order of symbol entry is $\text{Outside} \rightarrow \text{Radicand} \rightarrow \text{Radical}$. The probability of symbols being inserted in this order after 2 transitions when using MC$^2$ was approximately 0.16 versus 0.09 in Equation Editor, indicating users were nearly twice as likely to insert the expression in this order when using the 2-dimensional free-form model found in the MC$^2$ user interface.

Table 2. Square root transition probability matrix for expressions transcribed with MC$^2$ (left) and Microsoft Word Equation Editor (right)

| From     | Outside | Radical | Radicand |
|----------|---------|---------|----------|
| Outside  | 0.000   | 0.726   | 0.274    |
| Radical  | 0.232   | 0.000   | 0.768    |
| Radicand | 0.399   | 0.601   | 0.000    |

| From     | Outside | Radical | Radicand |
|----------|---------|---------|----------|
| Outside  | 0.000   | 0.758   | 0.242    |
| Radical  | 0.071   | 0.000   | 0.929    |
| Radicand | 0.633   | 0.367   | 0.000    |

3.2 Markov Chain Transition Probability Matrices for Fraction Structures

Table 3. Fraction transition probability matrix for expressions transcribed with MC$^2$ (top) and Microsoft Word Equation Editor (bottom)

| From                | To                  |
|---------------------|---------------------|
|                      | Numerator | Denominator | Fraction bar | Outside |
| Numerator            | 0.000      | 0.387       | 0.472        | 0.141    |
| Denominator          | 0.262      | 0.000       | 0.071        | 0.667    |
| Fraction bar         | 0.507      | 0.332       | 0.000        | 0.161    |
| Outside              | 0.506      | 0.018       | 0.476        | 0.000    |

| From                | To                  |
|---------------------|---------------------|
|                      | Numerator | Denominator | Fraction bar | Outside |
| Numerator            | 0.000      | 0.386       | 0.391        | 0.223    |
| Denominator          | 0.268      | 0.000       | 0.089        | 0.643    |
| Fraction bar         | 0.691      | 0.214       | 0.000        | 0.095    |
| Outside              | 0.281      | 0.018       | 0.701        | 0.000    |

Similar to the matrices exploring square roots in the previous section, the matrices in Table 3 show important differences in how users insert fractions when using the two models featured in the applications featured in this study. When using Microsoft Equation Editor, we can see from the transition matrices
that users are about 18% more likely to enter the numerator following insertion of the fraction bar or structure than they are when using MC². Furthermore, users were about 23% more likely to insert the numerator of a fraction following insertion of a symbol outside the fraction when using MC² than when using Equation Editor, where they are more likely to insert the fraction structure following insertion of a symbol outside of the fraction. These results show evidence that users are more likely to enter a fraction in the order, *Outside → Numerator → Fraction Bar → Denominator*, when using MC², with a probability of 0.08 after 3 transitions, than they were when using Equation Editor with a probability of 0.02 after three transitions. On the other hand, the transition matrices also show evidence that users are more likely to enter a fraction in the order, *Outside → Fraction Bar → Numerator → Denominator* when using Microsoft Equation Editor than they are when using MC². Again, we propose that this is due in part to differences in the flexibility of symbol insertion order when using these two applications. Generally, when using the structure based model featured in Equation Editor, an attempt to insert the fraction structure following insertion of the numerator will not result in the intended fraction and will require the user to delete the numerator and re-insert it into the numerator position of the fraction structure.

### 3.3 Markov Chain Transition Probability Matrices for Definite Integral Structures

**Table 4.** Definite integral transition probability matrix for expressions transcribed with MC² (top) and Microsoft Word Equation Editor (bottom)

| From         | Integral | Upper bound | Lower bound | Integrand |
|--------------|----------|-------------|-------------|-----------|
| Integral     | 0.000    | 0.536       | 0.428       | 0.036     |
| Upper bound  | 0.000    | 0.000       | 0.571       | 0.429     |
| Lower bound  | 0.000    | 0.462       | 0.000       | 0.538     |
| Integrand    | 1.000    | 0.000       | 0.000       | 0.000     |

| From         | Integral | Upper bound | Lower bound | Integrand |
|--------------|----------|-------------|-------------|-----------|
| Integral     | 0.000    | 1.000       | 0.000       | 0.000     |
| Upper bound  | 0.000    | 0.000       | 0.893       | 0.107     |
| Lower bound  | 0.036    | 0.036       | 0.000       | 0.928     |
| Integrand    | 0.500    | 0.000       | 0.500       | 0.000     |
The transition matrices for the order of insertion of symbols relative to definite integral structures, shown in Table 4, also show interesting results. Of primary interest is the conditional probability of the subsequent symbol inserted, given that the integral was inserted. In Microsoft Equation Editor, the upper bound of the definite integral was always the symbol inserted following insertion of the integral. However, in MC², the upper bound was inserted about 54% of the time following the insertion of the integral, and the lower bound was inserted about 43% of the time. Furthermore, users of MC² never inserted the integral following the insertion of the upper bound or lower bound. This may be a result of the different models used in the two applications. We suggest there may be an element of the 2-dimensional structural model used in Equation Editor which makes users more likely to enter in the upper bound of a definite integral following insertion of the integral itself. It should also be noted that there was only one definite integral present in the experimental stimuli and that the trials using MC² always occurred prior to those with Equation Editor. Therefore, these differences could be attributed to a learning or familiarity effect rather than a true difference in the two models.

### 3.4 Markov Chain Transition Probability Matrices for Summation Notation Structures

#### Table 5. Summation transition probability matrix for expressions transcribed with MC² (top) and Microsoft Word Equation Editor (bottom)

| From   | Sum | Upper index | Lower index | Inside | Outside |
|--------|-----|-------------|-------------|--------|---------|
| Sum    | 0.000 | 0.769 | 0.077 | 0.154 | 0.000 |
| Upper index | 0.000 | 0.000 | 0.917 | 0.083 | 0.000 |
| Lower index | 0.000 | 0.167 | 0.000 | 0.833 | 0.000 |
| Inside | 0.000 | 0.111 | 0.333 | 0.000 | 0.566 |
| Outside | 0.929 | 0.000 | 0.000 | 0.071 | 0.000 |

| From   | Sum | Upper index | Lower index | Inside | Outside |
|--------|-----|-------------|-------------|--------|---------|
| Sum    | 0.000 | 0.869 | 0.071 | 0.036 | 0.024 |
| Upper index | 0.000 | 0.000 | 0.823 | 0.107 | 0.000 |
| Lower index | 0.000 | 0.077 | 0.000 | 0.808 | 0.115 |
| Inside | 0.100 | 0.167 | 0.000 | 0.000 | 0.733 |
| Outside | 0.881 | 0.000 | 0.0595 | 0.0595 | 0.000 |
In Table 5 we see the transition matrices related to the summation notation for MC$^2$ and Microsoft Equation Editor respectively. These matrices show less drastic differences than those related to the other structures. One difference to note is the conditional probabilities of symbol insertion following insertion of the summation symbol. We can see that users are approximately 10% more likely to insert the upper index following the summation symbol when using Equation Editor than MC$^2$, similar to the integral approach explored above. Though not a large difference, this is interesting to note as there was only one summation notation present in the experimental stimuli. The relative similarity between these transition matrices gives more merit to the differences seen in the transition matrices for the definite integral structures, of which there was also only one present in the stimuli. Therefore, if there were learning effects which occurred, they were not necessarily consistent across all structures.

4 Limitations and Discussion

This study was designed to test and compare properties of the two featured models for representing mathematical structure in the applications used: a 2-dimensional structure-based model (Equation Editor) and a 2-dimensional free-form model (MC$^2$), rather than test which of the two programs performs better as a whole. Previously, Kang et al. [3] analyzed the completion time between these two software platforms, controlled for quality of expressions. It is worth noting that Microsoft Equation Editor has been adapted over the years to improve the user experience, and now includes many shortcuts which allow users to enter expressions in a more intuitive manner (including a limited subset of the $\LaTeX$ syntax). Participants were not given any instruction on the manner in which to use Equation Editor, and were therefore free to make use of these shortcuts if they wished. Most did not, most likely due to the fact that they were not aware that these existed, as the participants in this study were novices. To evaluate the overall effectiveness of the Equation Editor application in Microsoft Word, it might be more useful to make users aware that these shortcuts exist. However, the fact that most beginner level users were not able to discern that these shortcuts could be used may speak to the user experience of Equation Editor in itself.

There were a number of limitations present in this study. The most notable was the fact that all expressions were first transcribed using MC$^2$ followed by transcription with Microsoft Equation Editor. Therefore, for all 4 experimental stimuli, participants were more familiar with the expression to be transcribed during the Equation Editor trial than during the MC$^2$ trial. This could result in less interference with working memory caused by the Equation Editor user interface as the mathematical content would be internalized to a greater degree during these trials [8]. Furthermore, the mathematical expressions which were chosen as experimental stimuli were not chosen in a systematic fashion controlling for the complexity of each expression or the number of symbols or different types of mathematical structures found in each expression. This makes it difficult to fully attribute some of the effects found in this study to how the models
handle certain structures without considering some sort of interaction which may have occurred. Therefore, future research should be done to examine how the two featured models in these applications handle each of the structures, such as square roots, fractions, and exponents, in less complex expressions in which these structures are not featured in combination. Finally, the sample size for this study was rather small and although we control for the technical levels of students, we did not consider covariates while we constructed the Markov Chains, many of which may provide insight if sample sizes were sufficient.

Despite these limitations, this study presents a number of notable findings related to the user experience of typesetting applications which feature two different structural models for representation of mathematical notation. The Markov Chain analysis provides us with interesting insight into the process users are more likely to follow regarding the order of symbol insertion while transcribing mathematical content. Perhaps the most important finding is simply the fact that there is a difference in the order in which symbols are entered in both interfaces. If we assume that there is a natural order which individuals are more likely to follow when transcribing these expressions, regardless of the environment in which they are doing so, any sort of difference found here shows that either one, or possibly both of the models used in these interfaces forces users to need to adapt and adjust this order of insertion. Given the tendency found for users of the structure-based editor to insert the symbols in the upper position of an inserted structure (such as the numerator of a fraction) immediately following insertion of that structure, we recommend that editors using structure-based models place the cursor in this upper location immediately following insertion of the structure. In the Microsoft Equation Editor interface the cursor is placed outside and to the right of an inserted fraction structure. If the user wishes to insert the numerator next, they must navigate to the respective location using the arrow keys (pressing the left arrow twice) or click in the location using the mouse. Placing the cursor in the numerator position immediately following insertion of the fraction would increase the efficiency of the program as well as reduce the cognitive load, which could in turn decrease the working memory interference which has been shown to occur when digitally typesetting mathematics [8].

A number of differences in the conditional probabilities can likely be attributed to the reduced flexibility allowed by the 2-dimensional structure-based model found in Equation Editor. For example, when using the traditional implementation of this model in Word, a user cannot insert the numerator of a fraction before inserting the fraction bar itself. However, it is quite interesting that in all 2-dimensional structures with notation in which a symbol is located in an upper position and another one located in a lower position (fractions, definite integrals, summations) users were more likely when using Equation Editor to insert the symbol in the upper location following insertion of the structure itself (as compared to when using MC²). While the reasoning behind this effect is unclear, it merits further study. As all of the experimental stimuli featured expressions with multiple structures, it would be interesting to observe whether the tran-
sition matrices were similar for expressions containing only one of the featured structures, or if there is some level of interaction taking place.

5 Conclusion

In this paper we explored the user experience while engaged with two mathematical typesetting models and discovered several insights into the order symbols were entered when novice users transcribed mathematics. These insights suggest that further research in this direction could have important implications into the design of mathematical user interfaces, in particular interfaces for mathematical novices. Though findings should be interpreted with caution, the application of Markov transition matrices to symbol order entry in this research serves to guide future studies on the characteristics and usability of different models featured in WYSIWYG mathematical typesetting applications. Similar experiments should be conducted with more standardized experimental stimuli (mathematical expressions), varying the nested nature of the mathematical structure, to investigate how certain interactions might manifest themselves. Furthermore, for a more complete picture of how users see and interact with mathematical expression structure in software, future research might also include how novice users revise and fix input errors in an expression they have written. As more education moves online, developing and studying mathematical input models that are intuitive for novice users will likely become a more pressing issue. While this area has so far not received significant attention in the academic literature, we believe this paper makes an important first step in this direction.

References

1. Gozli, D.G., Pollanen, M., Reynolds, M.: The characteristics of writing environments for mathematics: behavioral consequences and implications for software design and usability. In: Carette, J., Dixon, L., Coen, C.S., Watt, S.M. (eds.) CICM 2009. LNCS (LNAI), vol. 5625, pp. 310–324. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02614-0_26

2. Jansen, A.R., Marriott, K., Yelland, G.W.: Parsing of algebraic expressions by experienced users of mathematics. Eur. J. Cogn. Psychol. 19(2), 286–320 (2007)

3. Kang, S., Pollanen, M., Damouras, S., Cater, B.: Mathematics classroom collaborator (MC2): technology for democratizing the classroom. In: Davenport, J.H., Kauers, M., Labahn, G., Urban, J. (eds.) ICMS 2018. LNCS, vol. 10931, pp. 280–288. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96418-8_33

4. Knauff, M., Nejasmic, J.: An efficiency comparison of document preparation systems used in academic research and development. PloS ONE 9(12), e115069 (2014)

5. Loch, B., Lowe, T.W., Mestel, B.D.: Master’s students’ perceptions of Microsoft Word for mathematical typesetting. Teach. Math. Its Appl. Int. J. IMA 34(2), 91–101 (2015)

6. 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). https://doi.org/10.1007/978-3-540-27818-4_22
7. Pollanen, M., Kang, S., Cater, B., Chen, Y., Lee, K.: $MC^2$: mathematics classroom collaborator. In: Proceedings of the Workshop on Mathematical User Interfaces (2017)

8. Quinby, F., Pollanen, M., Reynolds, M.G., Burr, W.S.: Effects of digitally typesetting mathematics on working memory. In: Harris, D., Li, W.-C. (eds.) Engineering Psychology and Cognitive Ergonomics. Mental Workload, Human Physiology, and Human Energy (LCAI), pp. 1–20. Springer (2020, in press)