

epsilonWriter: implementing new ideas for typing text and math

Jean-François Nicaud, Christophe Viudez

ARISTOD and Joseph Fourier University (LIG)

Grenoble, France

Jean-Francois.Nicaud@imag.fr

Christophe.Viudez@imag.fr

www.aristod.com

Abstract

We have developed a prototype for smoothly typing and editing text and math. Smoothly means that, for example, if the user types “Peter has $2/3$ of the age of Paul”, when “/” is typed, a fraction appears with the cursor at the denominator. A click places the cursor as well in a text word than in a mathematical expression (there is no special window to open) and the general principles of the editor of the Aplusix educational software have been used to implement smooth insertion, deletion, selection, copy, cut, past and drag&drop. An additional mechanism is the “Change” key which allows getting another result when available and which drastically decreases the number of buttons for choosing operators. For example, there is only one “sigma” button for sums; the different forms are reached with the “Change” key.

epsilonWriter is developed by the ARISTOD Company (www.aristod.com), a spin-off of the Joseph Fourier University. It is written in Java and will be available from July 2009 for several kinds of uses, the first ones being for scientific email and forums.

1. Introduction

Our current work is situated in the following of the “Natural Editing of Algebraic Expressions” [6] presented at MathUI2007. We are working on user interfaces with keyboard and mouse allowing writing and modifying texts and maths in a natural way, i.e., avoiding, as far as possible, rigid mechanisms described in [6] and observed in Microsoft Equation Editor [4], MathType [3], Scientific Word [8] or WIRIS [9], and taking into account mathematical properties. We use mechanisms implemented in the Aplusix educational software [1, 5] and new ideas presented in this paper.

Section 2 presents 6 principles for smooth editors of text and math, section 3 describes the epsilonWriter application and section 4 indicates how it will be distributed and the future work.

2. Principles for smooth editors of text and math

Here is a list of principles for typing texts and maths as smoothly as possible with a keyboard and a mouse.

1. *Use the same windows for text and maths.*

Often users of texts and maths editors are obliged to find a command for the insertion of a mathematical expression (can be 2 clicks) and to use double-click for modifying a mathematical expression. This can be improved: a simple click can place the insertion point in mathematical expressions as well as in texts.

2. Recognise some mathematical expressions.

Everybody knows that 2 is a mathematical expression. So, a smooth editor can do this as well and more generally can automatically interpret some beginning of typed expressions as mathematical expressions.

3. Use an algebraic syntax.

The foundation of algebraic domains is the existence of algebraic languages composed of operators and arguments. Instead of being based on this mathematical structure, most of the editors of algebraic expressions are based on a text&box structure which is the opposite of a smooth structure.

4. Be adequate and pleasant for keyboard-oriented users

Many advanced users are keyboard-oriented users. As far as possible, they prefer typing several keys on the keyboard instead of moving the hand to catch the mouse then search an adequate menu and an adequate item in this menu. Why not getting a fraction just by typing “/” or getting ω just by typing “omega”?

5. Be adequate and pleasant for mouse-oriented users

Some users like using the mouse, so it is a good point to have menus presenting all the operators and special characters. But something has to be found to avoid explosions: In Word 2007, there are 70 symbols in the “symbols of mathematical functions”, 4 fractions, 4 subscripts/exponents, 4 radicals, 9 integrals, 35 sums, products, unions, etc., 38 braces, square braces and parentheses, 30 functions (sin, cos, max, log, etc.), 20 over/above marks, 8 matrix, 4 multi-dots, which leads to 226 buttons, each one being reached with 2 or 3 clicks.

6. Allow smooth changes

The most unpleasant point with almost all editors is the complexity of operations like “keeping only the numerator of a fraction” where one would like to have this by suppressing the denominator, or “changing a symbol of integral to another, e.g., from simple to double integral”.

3. epsilonWriter

epsilonWriter is currently developed by the ARISTOD Company. About 80% of the code of the first version, to be distributed in July 2009, has been written and tested. This section describes the main technical points already implemented and some points which will be implemented for July 2009.

3.1. Only one window

epsilonWriter does not use a special window for maths. This choice is already present in Scientific Word [8] and Word 2007 [4]. A simple click places the insertion point as well in mathematical expressions as in texts. The insertion point is a blue bar in texts and a red bar in math. As in Scientific Word, the state of the current position is indicated in the toolbar and can be change par a click or by typing Alt-M (to math) or Alt-T (to text).

3.2. The “Change” action

An important idea implemented in epsilonWriter is the “Change” action. Frequently, in mathematical expressions, a user entry can be interpreted in different ways. In epsilonWriter, the user gets one

interpretation and can change it. The toolbar shows the number of different interpretations made by the system and the user can click on buttons or use the “Alt-C” key to move to other interpretations. This idea already exists for very specific actions in some editors (like “paste” in Word). It has been generalised in epsilonWriter.

Figure 1 illustrates the “Change” action for the insertion of “/”. Getting a fraction when typing “/” is something everybody would like to have. We do not find this in MathType, Scientific Word and Word 2007.

Figure 1. On the left, the insertion point is inside $3axy$ and “/” is type. One gets the second expression where a fraction is made with the surrounding arguments. Here, the operator is considered with a priority lower than “times”, so $3a$ goes to the numerator and xy to the denominator. The “Change” action provides successively the other expressions, and then goes back to the first interpretation. The second interpretation (third expression) gives the operator a priority higher than times. The third one inserts a fraction with empty arguments and the last one uses the “/” drawing.

3.3. Recognition of mathematical expressions

epsilonWriter recognises some mathematical expressions. Each time a new sequence of characters begins by a digit or a small sequence of characters continues with a mathematical operator, the expression is interpreted as a mathematical expression. This can be refused by the user with the “Change” action. When “space” is typed in a mathematical expression, epsilonWriter exits from the expression (there are exceptions described below). Figure 2 shows how mathematical expressions can be naturally inserted in texts. This feature does not exist in the other editors we tested.

Figure 2. In this typing, the user never does an action like “start an algebraic expression” or “exit of an algebraic expression”. First column: text is typed, the insertion point is blue; then 3 is typed and interpreted as maths, the insertion point is red; then “/” is typed providing a fraction with the insertion point at the denominator. Second column: 4 is typed, then “space” is typed, exiting from the fraction and allowing to type texts after the fraction.

3.3. “ToMath” and “ToText” transformations and separation principle

Like Scientific Word, epsilonWriter has commands “ToMath” and “ToText” to transform an expression into math representation of a text representation, but they run differently. epsilonWriter has a separation principle: there is at least a space or an end of line between two elements A and B being text words or mathematical expression. Scientific Word has not this property, you can have text at any position of a mathematical expression, see figure 3.

$$\frac{3x+2}{4x+2} \quad \frac{3x+2}{4x\text{hello}+2}$$

Figure 3. In Scientific Word, the insertion point is in a fraction. Clicking on the “ToText” button and typing “hello” provide the right expression where text and math are strangely mixed.

In epsilonWriter, “ToMath” transforms the word of the insertion point into math and “ToText” transforms the entire expression of the insertion point into text. When the space between a text word and a mathematical expression is deleted, the result is entirely text (when text is bigger) or math (when math is bigger) and can be changed with the change command, see figure 4.

these | $2n+3$ ways of...
these| $2n+3$ ways of...
these| $\times 2n+3$ ways of...

Figure 4. In epsilonWriter, the space between “these” and “ $2n+1$ ” is deleted by the user who gets a text form of the concatenation of the two parts. The “change” command provides a mathematical form.

3.4. Use of algebraic syntax

Managing mathematical expressions in editors has two contradictory aspects. On the one hand, one would like to have a control of the syntax. On the other hand, one would like to be able to type many sorts of things.

In epsilonWriter, mathematical expressions are quasi-well-formed expressions as in Aplusix. Operators are known by the system or indicated as custom operators by the user. Each operator has an arity (number of arguments) and missing arguments are displayed with a question mark. Some operators have optional arguments; this is the case of radicals which can have a 3 in the “V” part for a cubic root or nothing for a square root. In this case, no question mark is displayed, but it is possible however to insert a value. This is also the case of sums and integrals.

A special attention is paid to parentheses, braces, and square brackets which are only used in the following situations:

“{” is used for systems of equations/inequations with an “and” meaning, and “{}” are used for sets of elements.

“[]” are used for matrixes and “[” “]” for intervals.

“(” “)” are used for managing priorities of operators.

All these parentheses, braces, and square brackets, automatically get the right size, taking into account their arguments. Unbalanced parentheses are allowed for providing the capacity of changing the structure; they are displayed in red.

See figure 5 an example of deletion of parentheses.

$$3(x_i^2 + y_j^3) + z \quad 3x_i^2 + y_j^3 + z \quad 3(x_i^2 + y_j^3 + z$$

Figure 5. The user has the left expression and types “backspace”. He/she gets the second expression where the two parentheses are deleted. The “Change” action allows getting the right expression where only the left parenthesis is deleted.

3.5. For keyboard-oriented users

Each operator can be typed with the keyboard, some with a standard typing (like + - * /). It is very pleasant to get a superscript just by typing “^” and a subscript just by typing “_”. For other operators, a localised name is used, for example “root” in English and “racine” in French. Special characters, like Greek letters, can also be typed using a name like “omega”. In a near future, we will give the possibility to the user to choose the names of the actions and we will memorise this in a user profile. This is similar to the Latex notation [2] and the OpenOffice mathematical editor input [7], but in a WYSWYG way, without a specific window for typing commands and without a “\” prefix. Figure 6 shows how a matrix can be obtained with the keyboard.

$$\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} + \text{matrix} \quad \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} + \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Figure 6. The user types matrix. Just before typing the “x”, he/she gets the left expression. When the “x” is typed, the middle expression is obtained. It can be changed with the “Change” command to the right one.

In text, names of mathematical operators are recognised too and proposed by the change mechanism, see figure 7.

$$\text{asum} \quad a \sum ?$$

Figure 7. The insertion point is in a text area and the user types “asum”. He/she get the left figure which is the typed text but there is a “change” possibility which transforms the word in maths and interprets the sum. It is displayed on the right of the figure.

For right and left keyboard arrows, we chose to follow the semantic of right and left, i.e., “go to the right/left” except for right at the end of a line and left at the beginning of a line. Right and left arrows do not enter in superscript/subscript areas which are accessible with the up and down arrows. See figure 8 for an example and a comparison with other editors.

$$\sum_{i=1}^n x_i^2 \quad \sum_{i=1}^n x_i^2 \quad \sum_{i=1}^n x_i^2 \quad \sum_{i=1}^n x_i^2$$

Figure 8. The different positions of the insertion point obtained in epsilonWriter by typing the right arrow. Scientific Word goes in the superscript areas (“n” and “2”). MathType goes in the subscript “i” of “x” but not in the subscript “i=1” of the sum. Word 2007 goes everywhere (the two superscript and two subscript areas), forgetting the meaning of right and left.

For up and down Keyboard arrows, we chose to follow the semantic of up and down, i.e., “go up/down”. Up and down arrows allow going into superscript/subscript areas. MathType has this behaviour, Scientific Word has not, and Word 2007 has not, except for a special place of the insertion point.

For the “up” key, we generally chose (1) to go into the lower part of the upper expression/word and (2) to keep, as far as possible, the horizontal position of the insertion point of the first hit of up/down. See an example in figure 9 and a comparison with other systems.

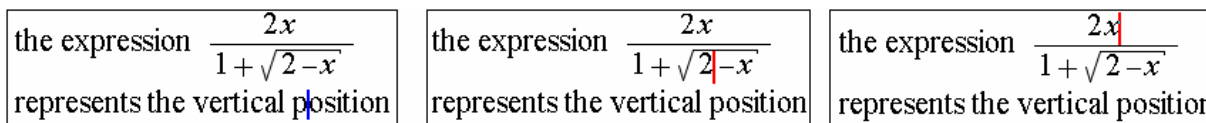


Figure 9. Two hits of the up arrow in epsilonWriter. MathType and Word 2007 have a similar behaviour for point (1) but do not verify point (2). Scientific Word only goes at the beginning or the end of the mathematical expression.

3.6. For mouse-oriented users

A toolbar provides all the characters and operators to the user. It is composed of a fix panel which contains the most often used items and several complementary panels which can be displayed one at a time or transformed into separated small windows, see figure 10.

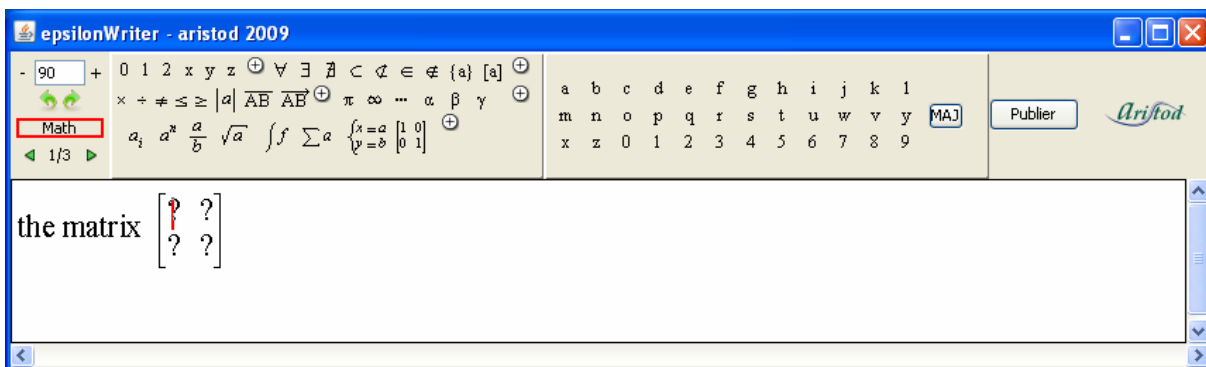


Figure 10. A screenshot of epsilonWriter. On the first panel: a zoom, the undo/redo button, the state Math/Text indication, the change button (here, we are on the first over 3 interpretation of the “matrix” input). The second panel contains the most “often used items” according to the designers and small buttons to choose the third panel.

3.7. The “space” and “return” keys in maths

In texts the space key asks to begin a new word, or to split a word, and the return key to begin a new paragraph or to split a paragraph. In mathematical expressions, split has often no meaning; this is the case when the insertion point is inside a square root or inside the numerator of a fraction. We have given a mathematical meaning to “return” for vertical structures which can be extended, i.e., systems of equations and matrixes, see figure 11. We have given a mathematical meaning to “space” for horizontal structures which can be extended, i.e., matrixes, see figure 12. In the other cases, they are interpreted by exiting the mathematical expression and being applied after.

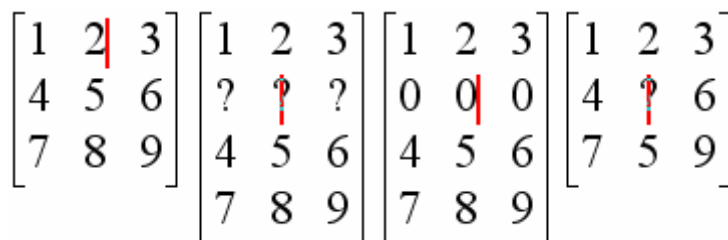


Figure 11. Interpretation of the “return” key in a matrix. The first interpretation (second matrix) is “insert an empty line”; the second, obtained by “change”, is “insert a line of zeros”; the third, obtained by “change”, is “insert an empty case and shift the rest of the column to the bottom”.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad \begin{bmatrix} 1 & ? & 2 & 3 \\ 4 & ? & 5 & 6 \\ 7 & ? & 8 & 9 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 2 & 3 \\ 4 & 0 & 5 & 6 \\ 7 & 0 & 8 & 9 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & ? & 5 \\ 7 & 8 & 9 \end{bmatrix}$$

Figure 12. Interpretation of the “space” key in a matrix. The first interpretation (second matrix) is “insert an empty column”; the second, obtained by “change”, is “insert a column of zeros”; the third, obtained by “change”, is “insert an empty case and shift the rest of the line to the right”;

These natural ways of interpreting “return” and “space” in mathematical expressions avoid fixed dimensions and the 4 well known menus “Add line/column above/below”.

3.8. Selection of an expression

We have implemented the same selection than in Aplusix [5]. Only sub-expressions can be selected, for example $x+3$ cannot be selected in $2x+3y$. Operations over a selection will be implemented soon: deletion, insertion of an operator, insertion of a non operator, cut, copy, paste and drag&drop.

3.9. Selection of an operator

epsilonWriter displays the structure of the expressions. More precisely, when the mouse cursor is over an operator, a rectangle (sometimes two) is drawn around the operator in red and rectangles are drawn around the arguments in blue, see figure 13. This help to understand the structure. When a structure is displayed, it is possible to fix the display by clicking on the operator, see figure 13. It is then possible to change the operator to another compatible operator or to delete the operator only, or delete with the arguments (delete will be implemented soon).

Figure 13. (1) an expression, (2) the mouse cursor is over the sum symbol, (3) a click on the sum symbol selects it, (4) another click on the sum symbol changes it, (5) a click at another position deselects.

4. Distribution and extensions of epsilonWriter

epsilonWriter is written in Java 1.4 and runs on many computers and operating systems.

The first uses are scientific forums and scientific email. Scientific forums will be installed on our website and customer websites and will use epsilonWriter for reading and writing messages. Messages will also be readable in a standard way mixing HTML for texts and images for mathematical expressions. Scientific email will be written on our website and finished messages will be redirected to the user mailer (in an HTML+image form) so that the user will benefit of his/her mailer environment to send the message. People will receive the HTML+image form on their usual mail environment and will have the possibility to answer with it (but mathematical formulas will not be editable) or using epsilonWriter (a link in the message will allow moving to the epsilonWriter website, where the original message will be provided in an editable form).

We plan to combine a free use of epsilonWriter for a non intense use (less than one hour writing per day) and a pay use by subscription for an intense use.

Several extensions are envisaged. Quizzes will probably be the first extension. A completion of the set of operators to be able to write all what can be written in sciences is envisaged. Arrays and graphics will be added. Reading and writing standards (MathML, OpenMath, Tex) is also a possibility. Running on PDAs and smartphones is another direction. So is the development of a plug-in form to be combined with pieces of software which will use epsilonWriter to get users' answers.

The development planning will be guided by commercial aspects, to be sure that the company can live from its sales.

5. References

[1] Aplusix. <http://aplustix.com>

[2] Latex. <http://www.latex-project.org/>

[3] MathType. <http://www.dessci.com/en/products/mathtype/>

[4] Microsoft Equation Editor. <http://office.microsoft.com/en-gb/word/HP051902471033.aspx>

[5] Nicaud, J.F., Bouhineau, D., Chaachoua H. (2004). Mixing Microworld and CAS Features for Building Computer Systems that Help Students to Learn Algebra. *International Journal of Computers for Mathematical Learning* 9, p. 169-211.

[6] Nicaud, J.F. (2007). Natural Editing of Algebraic Expressions. In proceedings of the MathUI workshop, Linz, Austria, 2007. <http://www.activemath.org/workshops/MathUI/07/>

[7] Openoffice <http://www.openoffice.org/product/math.html>

[8] Scientific Word <http://www.mackichan.com/>

[9] WIRIS editor. <http://www.wiris.com>