

Entering and Editing Mathematical Expressions on the Web

Wei Su^{1,2}, Paul S.Wang², Lian Li¹

¹Department of Computer Science, Lanzhou University, China

²Department of Computer Science, Kent State University, USA

Abstract

MathEdit is a Web-based open-source tool for entering and editing mathematical expressions. Implemented in Javascript, MathEdit works with IE and the Mozilla-family browsers and provides an easy-to-use GUI as well as an API for Javascript programs. MathEdit works with MathML and infix notations. By simply embedding the MathEdit tool, authors can add interactive entering, editing and manipulating mathematical expressions to their Web pages. MathEdit provides MathML-Content based and MathML-Presentation based editing as well as different visual navigation models for displayed expressions. The GUI appearance, menus and buttons, as well as other aspects of MathEdit can be customized through the API. Serialized forms (infix and MathML for now) of user created expressions can be transmitted to other programs for further processing.

1 Introduction

The Web is increasingly important in all kinds of directions. But entering and manipulating mathematical expressions are still difficult on the Web. We believe it is time to have a totally Web-based tool for the interactive entering and editing of mathematical expressions especially for applying the Web in of mathematics, science, and technology. It is also important to make such a tool easy to use and freely available.

The two distinct needs for mathematical expressions are to visually display mathematical formulas and to perform computations indicated by the formulas. The former is the *presentation* aspect whereas the latter is the *semantics* aspect of a mathematical expression.

The Mathematical Markup Language (MathML) [22], a W3C standard, supports both a *presentation encoding* and a *content encoding* for these different purposes.

In the past decade, many companies and research institutes have developed mathematical expression editors [5, 6, 8, 10, 12, 13, 18, 19, 23]. These editors usually aim either to capture the meaning or to describe the visual appearance. The Amaya [1], LyX [9], TeXmacs [16] and MathType [3] using MathML Presentation [22], LaTeX [7], or native formats to store expressions are suitable for describing the expression appearance. Most Computer Algebra Systems, such as Maple [2] and Maxima [4], and a few independent editors, such as MathEX [14], use infix and MathML Content [22] to capture the meaning of expressions.

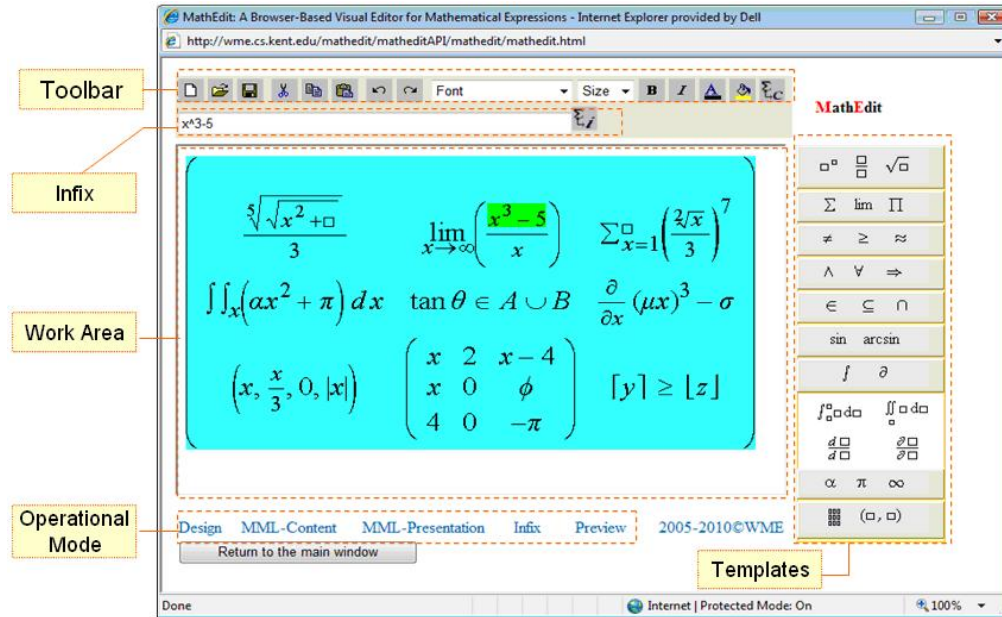


Figure 1: MathEdit Authoring Environment

Kent State University and Lanzhou University are jointly developing an editor for mathematical expression, called MathEdit [15, 20, 21], to satisfy both the need for visual display and the need for expression processing.

MathEdit, a Web-based visual interactive mathematical expression editor, aims to provide an integrated solution of on-line entering and editing of mathematical expressions. Figure 1 shows MathEdit authoring environment for end user (boxes with black lettering are part of the illustration). MathEdit API can be used for authors of Web-based systems to include MathEdit into their interactive Web page. By customizing by MathEdit API, MathEdit supports Content-based editing and Presentation-based editing for the two purpose of entering mathematical expression mention above. Implemented in JavaScript, MathEdit is a totally Web-based system, which can run in browsers both on IE and Firefox

The paper will introduce the implementation, editing method, navigation, and API of MathEdit. Section 2 will address the general editing method of MathEdit and the purpose and particular operations in Content-based editing. Section 3 will emphasize on Presentation-based editing and what is different between the two editing methods. Section 4 will describe the customization features and API library of MathEdit. We also give two examples of applying MathEdit in section 4.

2 Content-based Editing

Content-based editing enables the user to enter/modify well-formed expressions that represent meaningful mathematical operations. Internally, MathEdit uses an operation-operand tree structure to support the editing. Once created, the expression can be output in different formats:

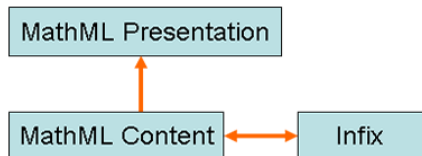


Figure 2: The Relationship of MathML Content, Presentation and Infix in MathEdit



Figure 3: Entering Binary Operator in MathEdit Content-based Editing

MathML Content code, Presentation code, and infix code. The infix string is a format understood by most computer algebra and computer geometry systems. Figure 2 shows the relationship among the three kinds of code. MathML Content format is central for MathEdit content-based editing which can be converted to MathML Presentation for display in interactive editing, and interconvert with infix, for conveniently entering expressions. In the design of MathEdit, we try to offer a conformable editing method for both Content-based editing and Presentation-based editing. But still lots of operation and navigation behaviors can not be kept completely the same. First some general information on expression editing and navigation in MathEdit.

2.1 Expression Editing

It is essential to provide convenient and intuitive operations to insert and remove the desired structures, including sub-expressions, operators, and characters. MathEdit for content-based editing supports several forms of user input, infix, MathML and template menus. The common approach is to use graphical templates that can be selected to enter particular kinds of mathematical expressions such as fractions, square roots, powers, and so on. Infix is convenient for users who are familiar with the notation. Two infix strings, one for the current sub-expression and the other for the whole expression, will be created and displayed synchronously with interactive template-based editing. A user can also edit either infix box directly to replace the current sub-expression or the entire expression. Infix editing can also solve some problems which are hard to implement in template-based editing, such as removing cube root or changing the division symbol in a fraction to a plus sign. Advanced users can also view, enter, and edit MathML content source code directly.

Content-based editing adopted many measures to keep the expression well-formed. For example, when a user, after selecting an operand, enters a binary operator such as $-$ (minus) the second operand of the binary operator will be added as a *placeholder* automatically (See Figure 3). Figure 4 also shows another example where a *placeholder* replaces a deleted sub-expression. The template menu also provide common expressions with explicit semantics for users to choose. Direct navigation of the content expression tree further ensures meaningful editing that preserves semantics.



Figure 4: Deleting Operation in Content-based Editing

2.2 Navigation

An important user interface aspect of mathematical expression editing is navigating to the precise point within the expression where an editing operation is to take place. The navigation method is largely the same in MathEdit Content-based editing and Presentation-based editing. Editing operations, such as insert, edit, delete, covert, and replace, are relative to the *current node* (a node on the MathML Content tree in Content-based editing and a node on the MathML Presentation tree in Presentation-based editing). *Current node expression* (the current expression) is highlighted visually and visual navigation refers to moving the current expression to different positions in the expression being edited. Convenient visual navigation is important and a user has multiple ways to visually navigate the displayed expression.

The four arrow keys are used for the basic navigation. They move the current sub-expression up to the parent node, down to the first child, or left/right to sibling nodes. Because arrow key navigation is based on the hidden internal tree structure, it can be non-intuitive to the average user. MathEdit also supports a systematic traversal of the entire expression so the user has a way of reaching any node on the tree. By pressing the `PageDown` key, the current node is moved in a traversal sequence defined by DFS (depth-first search) algorithm. The `PageUp` key, on the other hand, provides the inverse-orient traversal. To make navigation more intuitive, MathEdit also allows user to use the mouse click to select the current leaf node. Combining mouse clicks with the arrow keys can make navigation more convenient.

In Content-based editing the current sub-expression is always a well-formed mathematical expression. In other words the single operator can't be selected in Content-based editing. There are both advantages and disadvantages for this navigation way. One of the advantages is the user can always see and edit the infix and MathML content code of the well-formed sub-expression. And the other advantage is to avoid the user making the expression ill-formed by, say, deleting the $+$ sign in the formula $\frac{3}{4} + 5$. But the disadvantage is the user can't modify the operator to a $-$, say, directly.

3 Presentation-based Editing

Presentation-based editing enables users to enter arbitrary expressions that may or may not have widely accepted mathematical meaning or may not be mathematical at all. Different from Content-based editing, MathEdit Presentation-editing adopted MathML Presentation markup as its internal representation. And each edit operation, mouse click, or keyboard input, basically adjusts the DOM tree of MathML Presentation markup. Because Presentation-based editing only describes the display layout of an expression, it allows the user to enter any symbol at any position within an expression as long as the expression can still be represented by the MathML Presentation markup internally.

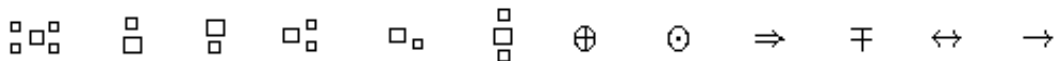


Figure 5: Some Templates in Presentation-based Editing



Figure 6: Entering Binary Operator in MathEdit Presentation-based Editing

For users, most input and navigation methods under Presentation-based editing are the same as under Content-based editing. Now, let's focus on the features particular to Presentation-based editing.

Two editing methods, template and MathML Presentation code, are supported in the Presentation-based editing of MathEdit. For the intention of Presentation-based editing is to describe the layout or appearance of an expression, MathEdit provides more templates for users to enter expressions with complex appearances (See Figure 5). There is also no need for using placeholders as operands to ensure the expression well-formed. Figure 6 shows the results with the same input operations as in Figure 3. Because the infix notation is inherently not related to visual layout, Presentation-based editing does not support infix input.

Presentation-based editing also adopt arrow keys, PageDown key and mouse to implement basic navigation, traversal navigation and click navigation. Selecting an operator as the *current node* is allowed.

Table 1 lists the differences between Content-based and Presentation-based editing in MathEdit. Offering two different editing modes has advantages and disadvantages. The advantage obviously is increased editing capabilities for users. The disadvantage is having to learn the difference between the two and to know when to use each method.

Table 1: The Difference between Content-based Editing and Presentation-based Editing

Item	Content-based Editing	Presentation-based Editing
Template	Yes	Yes (More Layout Templates)
Infix	Yes	No
MathML Content	Yes	No
MathML Presentation	Yes (Display)	Yes (Display and Editing)
Mouse Click	Yes	Yes
Arrow Key	Yes	Yes
PageDown key	Yes	Yes

4 Customization and MathEdit API

In designing any software, it is important to carefully balance ease of use, power and functionality, and the needs of different user groups. The key in MathEdit to solve this usability vs. complexity issue is to make the tool easily extensible to add functionalities and customizable to suit the needs of different user groups. The MathEdit API, supported by a set of JavaScript functions, allows applications to create an editor instance, sets its editing ways, configure its template palette and other GUI features, and set/retrieve the MathML code and infix it contains.

In some actual Web-based systems, Content-based editing and Presentation-based editing may be required in one system. For example, an on-line education system may need Content-based expression editing for an interactive lesson page, and Presentation-based expression entering for a chat page. MathEdit can provide both editing modes with similar user interfaces for end users that can be customized by uniform API functions. Through the MathEdit API, Web authors can also define an editor instance, which includes templates of Content-based editing and Presentation-based editing together in one palette. As mentioned above, Content-based editing also generates MathML Presentation markup. Thus the end user can use default Content-based editing to enter most expressions that can be displayed on the Web and passed to code or external systems for further computation. Of course, the editor instance can automatically switch to Presentation-based editing when user clicks on Presentation-based template to enter some special visual structure expressions. The switching is a one-way operation. A presentation expression can't be switched back to Content-based editing.

Figure 7 and Figure 8 show two instances of applying MathEdit. In Figure 7 MathPASS, a Web-based mathematical assignment system being used by Department of Mathematics, Kent State University, took MathEdit as its answer entering tool. The entered answer will be checked by computing in Maxima. By customizing through MathEdit API, a MathEdit instance with compact user interface and Content-based editing was embedded in the assignment system.

MathBoard shown in Figure 8 is another applying sample of MathEdit. MathBoard is an ongoing sub-project of Web-based Mathematics Education (WME) [11]. And it aims to create a mathematical bulletin board system (BBS) on the Web. Teachers and students can use MathBoard to post discussion mathematical topics, questions, responses, and comments. MathBoard has ability of on-line mathematical expressions editing powered by MathEdit Presentation editing.

The MathEdit API also allows the application to choose between a pop-up editor window, an in-page editing window, a simple infix editing box, or even an instance hidden from user view.

5 Future Work

MathEdit offered basic Content-based editing and Presentation-based editing in order to satisfy the requirement of editing mathematical expressions on the Web. The next stages of our work are intended to reduce the differences of operations between two editing ways and make the two ways able to integrate seamless. On the other hand the improvement of ability for capturing semantic meaning in Content-based editing and describing more layout appearance in Presentation-based editing will be a long term work of MathEdit. More formats of mathematical expression, such as OpenMath [17], LaTeX will be supported in MathEdit.

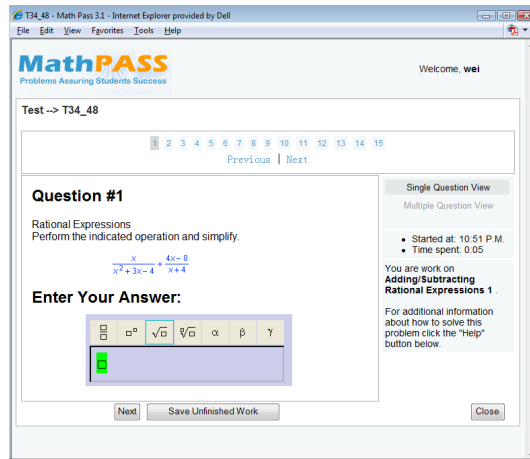


Figure 7: MathPASS, an Applying Instance of MathEdit

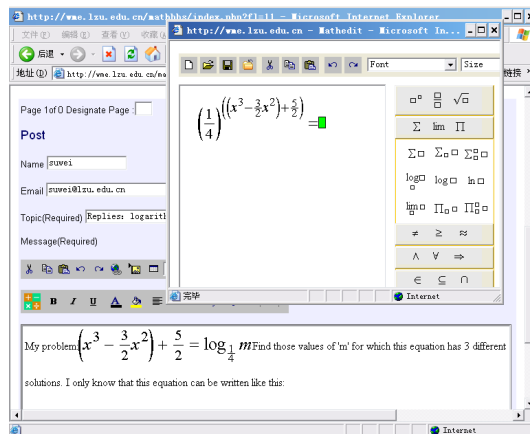


Figure 8: MathBoard, an Applying Instance of MathEdit

6 Acknowledgments

We'd like to thank Dr. Xiao Zou of Kent State University for making available his mathematics encoding converter for use in MathEdit.

The material reported here is based upon work supported in part by the National Science Foundation of USA under Grant CCR-0201772, the National Natural Science Foundation of China under Grant 90612016, and the Guangdong Provincial Government and Chinese Ministry of Education under Grant 0712226100097. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

- [1] Amaya Homepage <http://www.w3.org/Amaya/>
- [2] Document of Maple <http://www.maplesoft.com/Products/Maple/>
- [3] Document of MathType <http://www.dessci.com/en/products/mathtype>
- [4] Document of Maxima <http://maxima.sourceforge.net/>
- [5] Giorgi Gogvadze, Jan Tsigler, *Authoring Interactive Exercises in ActiveMath*, Proceeding of Mathematical User-Interfaces Workshop 2007, Schloss Hagenberg, Linz, Austria, June 27 2007.
- [6] Kehinde Alabi, Generation, *documentation and presentation of mathematical equations and symbolic scientific expressions using pure HTML and CSS*, Proceedings of the 16th international conference on World Wide Web, Banff, Alberta, Canada, May 08-12, 2007.
- [7] LaTeX documentation <http://www.latex-project.org/guides/>
- [8] Luca Padovani and Riccardo Solmi, *An Investigation on the Dynamics of Direct-Manipulation Editors for Mathematics*, MKM 2004, LNCS 3119, pp. 302C316, 2004.
- [9] LyX <http://www.lyx.org/>
- [10] Mathmled <http://www.newmexico.mackichan.com/MathML/mathmled.htm>
- [11] P. Wang, M. Mikusa, S. Al-shomrani, D. Chiu, X. Lai, and X. Zou, *Features and Advantages of WME: a Web-based Mathematics Education System*, IEEE SoutheastCon 2005.
- [12] Paul Libbrecht and Dominik Jednoralski *Drag-and-drop of Formula from a Browser*, Proceeding of Mathematical User-Interfaces Workshop 2006, St Anne's Manor, Workingham, United Kingdom, 2006 Aug 10th.
- [13] Samuel S. Dooley, *Editing Mathematical Content and Presentation Markup in Interactive Mathematical Documents*, Proceedings of ISSAC, 2002.
- [14] Samuel S. Dooley, *MathEX: A Direct-Manipulation Structural Editor for Compound XML Documents*, Proceeding of Mathematical User-Interfaces Workshop 2007, Schloss Hagenberg, Linz, Austria, June 27 2007.
- [15] Su Wei, Paul S. Wang, Li Lian, *An On-line MathML Editing Tool for Web Applications*, Proceeding of International Multi-Symposiums on Computer and Computational Sciences 2007 (IMSCCS07) The University of Iowa, Iowa City, Iowa, USA, August, 2007.
- [16] TeXmacs <http://www.texmacs.org/>
- [17] The OpenMath Standard 2.0 Draft <http://www.openmath.org/>
- [18] The W3C MathML software list <http://www.w3.org/Math/Software/>

- [19] WebEQ Documentation <http://www.dessci.com/en/products/webeq>
- [20] Wei Su, Paul S. Wang, Lian Li, Guanyu Li, and Yanjuan Zhao, *MathEdit, A Browser-based Visual Mathematics Expression Editor*, Proceedings of ATCM 2006, Hong Kong, China, 2006.
- [21] Wei Su, Paul S. Wang, Lian Li, *Features and Advantages of MathEdit, a Web-Based Visual Interactive Editor for Mathematical Expressions*, Presentation in The 15th International Conference on Learning, University of Illinois, Chicago, USA, June 2-6 2008.
- [22] W3C Math <http://www.w3.org/Math>
- [23] Y. Doleh and P. S. Wang, *A System Independent User Interface for an Integrated Scientific Computing Environment*, Proceedings of the ISSAC'90, Addison-Wesley (ISBN 0-201-54892-5), Aug. 1990, pp. 88-95.