# Browser-based Mathematical Formula Editing for The Web

Wei Su[a,b], Paul S. Wang[b], Lian Li[a]

[a] *Department of Computer Science, Lanzhou University, China*
[b] *Department of Computer Science, Kent State University, USA*

**Abstract**

MathEdit is an interactive tool for creating and editing mathematical expressions on the Web. MathEdit is an open-source program implemented in standard XHTML and JavaScript to run in regular browsers. The tool supports both WYSIWYG editing and command-line editing operations. MathML is the primary internal representation for MathEdit. But other formats such as infix, OpenMath, and LaTeX are also supported. A well-defined API enables interactions between the editor and its hosting Web page. Preference settings and customizations allow MathEdit to fit in different application environments. The design, implementation, and application of MathEdit are presented together with a comparison with several other mathematical expression editors.

*Key words:*
MathML, OpenMath, LaTeX, Mathematical Formula Editing, Visual Editing, Mathematical Expression

## 1. Introduction and Overview

The Web has come a long way since the early 1990's and has become ubiquitous in modern life. However, dealing with mathematical formulas is still difficult on the Web. With the advent of MathML [30] as a standard representation format for rendering (MathML Presentation) and semantics (MathML Content) and browsers support for MathML, either natively or via a plug-in [15], the foundation has been laid for mathematics on the Web. To realize this goal, we still need a Web-based tool to allow users and Web pages to interact with mathematical formulas in a convenient and natural way.

The *Web-based Mathematics Education* (WME) project [17] at the *Institute for Computational Mathematics* (ICM/Kent) was started in the mid 1990's to build an innovative on-Web mathematics education environment for middle school teachers and students. As part of WME, work also began on an interactive visual editor for mathematical expressions that runs in standard Web browsers and works with standard mathematics representations such as MathML, infix, and LaTeX [9].

In this direction, Kent State University and Lanzhou University jointly have developed MathEdit [22, 27, 28, 29], an open-source tool running in standard browsers for entering and editing mathematical expressions for the Web. MathEdit allows users to create and edit mathematical expressions with a convenient and intuitive graphical user interface (GUI) as well as an efficient command-line environment with character-string input. Using well-defined API functions, MathEdit can also be embedded in the interactive Web application systems by authors to create mathematical expressions.

Figure 1 shows the end-user view of the MathEdit architecture. Through the GUI and character-string input box, user actions, mouse clicks, and keyboard input, are treated as commands. Commands invoke JavaScript [21] functions that operate on HTML and MathML DOM [3] trees to support editing and visual navigation of mathematical expressions. MathML Presentation and Content codes are basic to the internal operations of MathEdit. But MathEdit also provides format conversion that can convert the format for the expressions among MathML, OpenMath [24], LaTeX, and infix. Each edit operation basically adjusts the DOM tree of MathML markup kept internally for the mathematical expression being constructed or edited. The effect of each editing operation is reflected in the visual display immediately.

MathEdit provides different user input modes, convenient API for the host Web page. Preference settings and customizations at the user and program levels make it possible to use MathEdit for different purposes and at different levels of mathematics. In the next sections, we will first discuss how MathEdit is applied then discuss several important aspects of the design and implementation of MathEdit.

## 2. Applications of MathEdit

MathEdit can be embedded in any Web page to enable users to easily input and edit mathematical expressions. Such expressions may be displayed and processed further as mathematical formulas. Before describing various

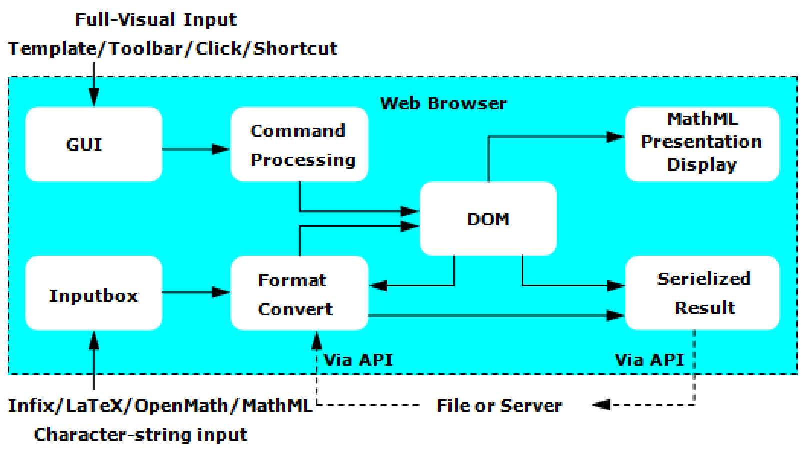


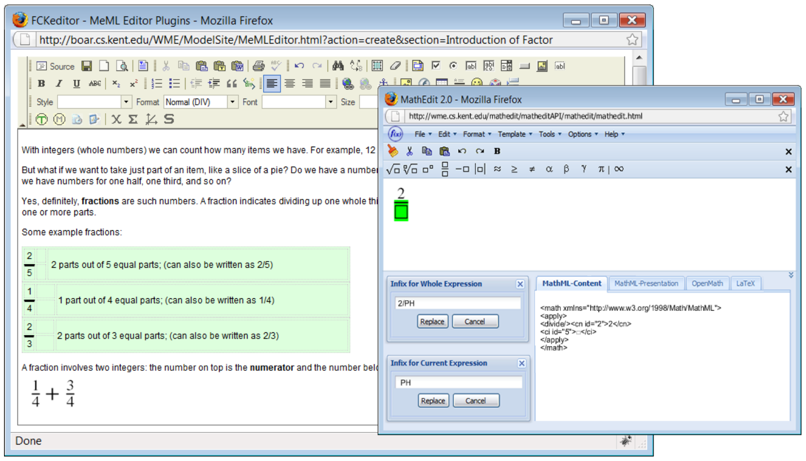
Figure 1: The architecture of MathEdit

design and implementation aspects of MathEdit, it is good to first see some actual applications of MathEdit.

As one can easily image, MathEdit is applied extensively in WME: to collect formula input in questions, exercises and tests, to support interactions between formulas and geometrical objects or plots of curves and surfaces, and to make interactive, step-by-step, solutions of equations possible.

Figure 2 shows another example: adding or changing mathematical expressions in a WME lesson page. Clicking the MathEdit icon pops up a MathEdit window for entering a new mathematical expression or editing an existing one which the user had selected from the WME lesson before invoking MathEdit. When the user is done, the formula created is returned (in MathML) to the lesson page to replace the existing expression or to be inserted anywhere in the lesson page. Every returned mathematical expression from MathEdit is given an HTML element `ID`. The HTML element `ID` of any MathML expression can be passed to MathEdit through its API for editing or format conversion. For example, a Web page may obtain infix, MathML, or OpenMath of a formula by passing its `ID` to MathEdit and then use the result to draw mathematical curves and surfaces or perform other mathematical operations.

Another significant application of MathEdit is in MathPASS (Figure 3), an interactive drill-and-practice system for remedial math classes at Kent State University. These classes regularly enroll more than 800 students and



FCKeditor - MeML Editor Plugins - Mozilla Firefox

http://boar.cs.kent.edu/WME/ModelSite/MeMLEditor.html?action=create&section=Introduction of Factor

Style: Format: Normal (DIV) Font: Size:

With integers (whole numbers) we can count how many items we have. For example, 12

But what if we want to take just part of an item, like a slice of a pie? Do we have a number
we have numbers for one half, one third, and so on?

Yes, definitely, **fractions** are such numbers. A fraction indicates dividing up one whole thi
one or more parts.

Some example fractions:

| | |
|---|---|
| $\frac{2}{5}$ | 2 parts out of 5 equal parts; (can also be written as 2/5) |
| $\frac{1}{4}$ | 1 part out of 4 equal parts; (can also be written as 1/4) |
| $\frac{2}{3}$ | 2 parts out of 3 equal parts; (can also be written as 2/3) |

A fraction involves two integers: the number on top is the **numerator** and the number belo

$\frac{1}{4} + \frac{3}{4}$

Done

MathEdit 2.0 - Mozilla Firefox

http://wme.cs.kent.edu/mathedit/matheditAPI/mathedit/mathedit.html

Infix for Whole Expression: 2/PH

Infix for Current Expression: PH

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
<apply>
<divide/><cn id="2">2</cn>
<ci id="5">□</ci>
</apply>
</math>
```

Figure 2: WME lesson author tool

involve multiple instructors. Teachers can easily create algebra questions in MathPASS with the built-in MathEdit through infix input. Presentation MathML code for rendering on the Web (Figure 4) will be generated automatically. Students using MathPASS can choose between two answer input styles offered by MathEdit: infix string input or GUI-based WYSIWYG input. Through simple MathEdit API calls, MathPASS provides a button for the students to select or change the entry methods. After entering an answer via MathEdit, a student clicks the `check answer` button which triggers JavaScript in the Web page to obtain the string representation of the formula created via the MathEdit API. The formula string then becomes part of a POST request to an answer checking program.

## 3. Editing Styles and Output Formats

It is essential for any mathematical expression editor to provide convenient and intuitive operations to insert and delete mathematical constructs including numbers, constants, variables, operators, functions and sub-expressions. Two editing styles are supported: full-visual input and character-string input. In full-visual input style the expression is edited by directly manipulating and navigating its on-screen display. In character-string input style editing can be done via infix or some other string representations of the expression.

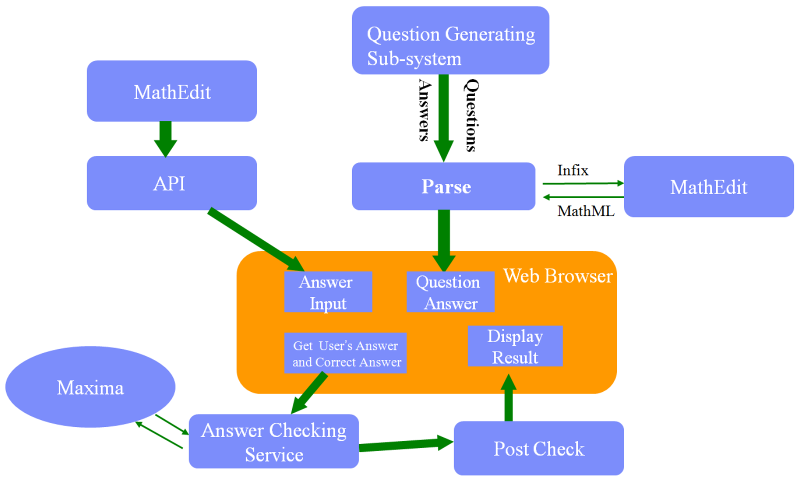
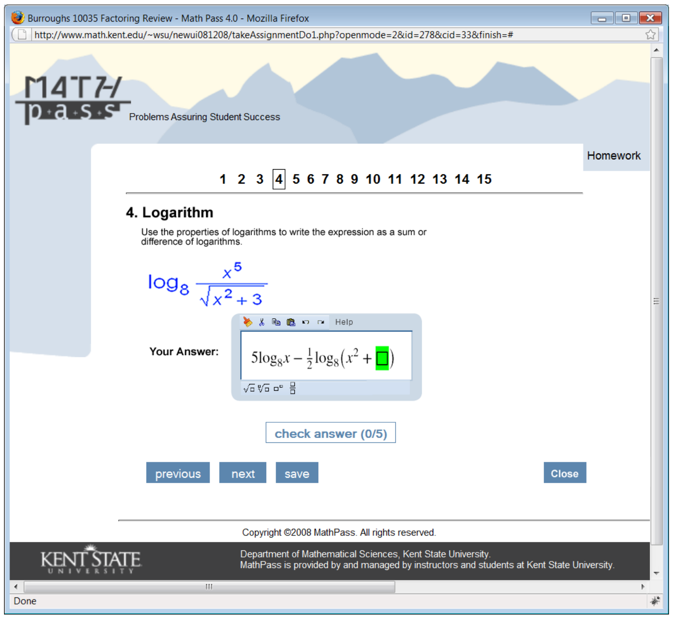


Figure 3: The architecture of MathPASS


Burroughs 10035 Factoring Review - Math Pass 4.0 - Mozilla Firefox

http://www.math.kent.edu/~wsu/newui081208/takeAssignmentDo1.php?openmode=2&id=278&cid=33&finish=#

Problems Assuring Student Success

Homework

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

**4. Logarithm**

Use the properties of logarithms to write the expression as a sum or difference of logarithms.

$$\log_8 \frac{x^5}{\sqrt{x^2+3}}$$

**Your Answer:** $5\log_8 x - \frac{1}{2}\log_8(x^2 + \square)$

Copyright ©2008 MathPass. All rights reserved.

Department of Mathematical Sciences, Kent State University.
MathPass is provided by and managed by instructors and students at Kent State University.

Done

Figure 4: An screen shot of MathPASS

### *3.1. Full-Visual Input Style*

In full-visual input, graphical templates of various mathematical constructs are used to enter expressions. The displayed expression grows and shrinks as editing continues on a real-time basis. Templates available include fractions, square roots, powers, and so on and can be customized for particular levels or areas of mathematics. Numbers, symbols, and simple operators (`+`, `-`, `*`, and `/`) can be entered from the keyboard. In full-visual input, authors edit in-place a mathematical expression as it is formatted and displayed on the screen in the traditional 2D textbook form. The actual display is done by the MathML renderer of the particular Web browser. Editing and displaying occur simultaneously and the expression is reformatted immediately after every modification.

An important user interface aspect of full-visual input is navigating to the precise point within the expression where an editing operation is to take place. 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: *basic navigation*, *traversal navigation*, and *cursor navigation*.

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.

The other important aspect of full-visual input is entering and editing the expressions. In MathEdit the tokens (digit, letter, operator and sub-expression) can be entered by keyboard or mouse. The command processor analyzes the previous token and the input token incrementally as it is received based on series *editing rule*. These *editing rule* could make sure the editing expression in computer is convenient and coherent with writing an expression in paper by pen. Based the analysis result, the command processor module will adjust the DOM of MathML.

## *3.2. Character-String Input Style*

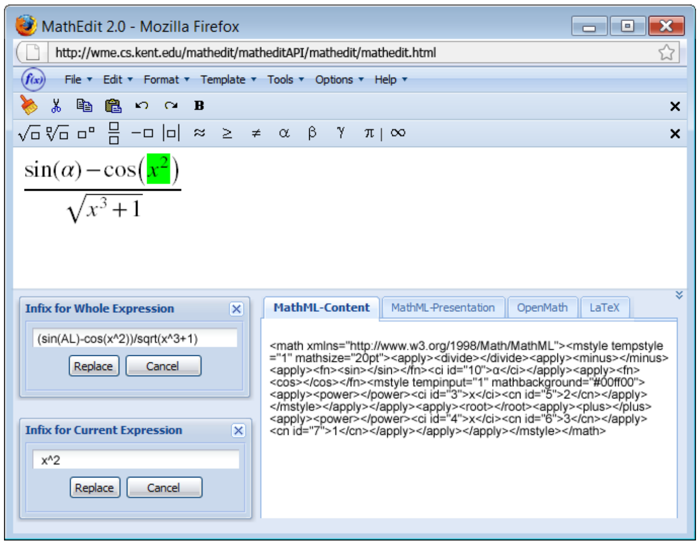
Character-string input provides a natural and unambiguous editing method in MathEdit. Multiple character-string input formats including infix, LaTeX, MathML, and OpenMath give user more ways to enter mathematical expressions. Infix and LaTeX are two common mathematical expression encoding formats, which are compact and easy to understand. Users familiar with LaTeX may like use that notation to enter expressions. LaTeX is a de facto standard for research publications but not widely used in schools or colleges. Advanced users can also view, enter, and edit MathML and OpenMath source code directly.

During the character-string input, the 2D mathematical expression can be shown immediately with each user modification (synchronous) or only after the user clicks a button (asynchronous). In synchronous mode, the MathEdit format conversion module monitors user input: analyzing the input syntax, converting the input to MathML Presentation, and rendering it on the screen. In asynchronous mode, all this work is only performed on the input string after the button click.

By providing multiple input styles, user interactions in MathEdit can be more flexible and user-friendly. Both character-string and full-visual input styles are available to the user all the time. This means the user can use one or the other input style as editing continues. In actual usage, this hybrid input method proves most effective. In Figure 5, two infix strings, one for the current sub-expression and the other for the whole expression, were created and displayed via full-visual input. Now a user can edit either infix box directly to replace the current sub-expression or the entire expression. The character-string input method can also solve some problems which are hard to implement in full-visual input, such as removing cube root or changing the division symbol in a fraction to a plus sign.

## *3.3. Standard Infix Format*

In all the character-sting input formats, infix is the most readable and efficient for entering mathematical expressions. Infix is also most widely used


MathEdit 2.0 - Mozilla Firefox
http://wme.cs.kent.edu/mathedit/matheditAPI/mathedit/mathedit.html
File
Edit
Format
Template
Tools
Options
Help
Infix for Whole Expression
Replace
Cancel
Infix for Current Expression
Replace
Cancel
MathML-Content
MathML-Presentation
OpenMath
LaTeX

$$\frac{\sin(\alpha)-\cos(x^2)}{\sqrt{x^3+1}}$$

```
(sin(AL)-cos(x^2))/sqrt(x^3+1)
```

```
x^2
```

```
<math xmlns="http://www.w3.org/1998/Math/MathML"><mstyle tempstyle
="1" mathsize="20pt"><apply><divide></divide><apply><minus></minus>
<apply><fn><sin></sin></fn><ci id="10">α</ci></apply><apply><fn>
<cos></cos></fn><mstyle tempinput="1" mathbackground="#00ff00">
<apply><power></power><ci id="3">x</ci><cn id="5">2</cn></apply>
</mstyle></apply></apply><apply><root></root><apply><plus></plus>
<apply><power></power><ci id="4">x</ci><cn id="6">3</cn></apply>
<cn id="7">1</cn></apply></apply></apply></mstyle></math>
```

Figure 5: A customized user interface for MathEdit

in computing systems such as computer algebra systems [4, 5, 7], general-purpose programming languages, and electronic calculators. Microsoft Office 2007 also begins to use a plain-text version of infix notation for entering mathematical expressions in Unicode [16].

At Kent State University, a standard infix format for mathematical expressions was created to let users enter expressions through ASCII keyboard efficiently. Unlike being concerned with presentation in [2, 16], the standard infix format represents the semantic content of mathematical expressions with proper typographical display. The standard infix of representations can easily inter-operate with math-oriented programs.

Table 1 lists the five main syntax rules of standard infix format. In basic operands, the symbols may be one or more characters or subscripted variables such as `x[1]`, `y[2]` which will be displayed as $x_1$ and $y_2$. Our format also defines special all-caps symbols for Greek alphabets and special symbols such as $\infty$ (`INF`) and $\varnothing$ (`NO`). Operators observe the usual precedence and associative rules. Sub-expressions could be grouped by parentheses. Figure 5 shows a typical example of $\frac{sin(\alpha)-cos(x^2)}{\sqrt{x^3+1}}$ in standard infix format as `(sin(AL)-cos(x^2))/sqrt(x^3+1)`.

Table 1: The syntax rules of standard infix format

| | |
|---|---|
| `o::= numbers \| symbols \| constants` | *basic operands* |
| `u::= + \| - \| not \| ...` | *unary operators* |
| `b::= + \| - \| * \| / \| ^ \| % \| > \| in \| => \| and \|...` | *binary operators* |
| `f::= sqrt(e) \| root(e,e) \| sin(e) \| diff(e,e) \|...` | *functions* |
| `e::= o \| f \| ue \| ebe \| (e)` | *infix expression* |

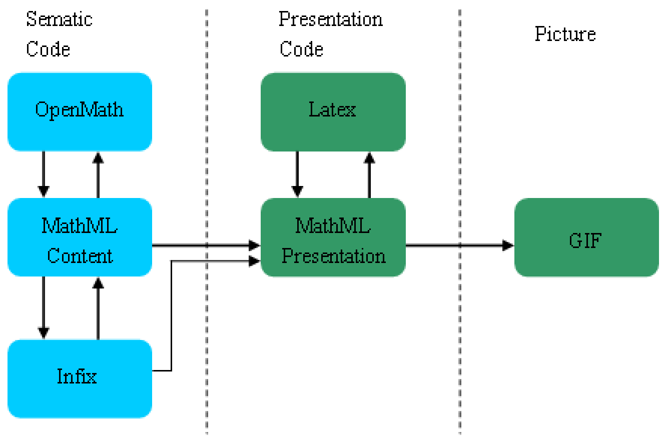


Figure 6: The relationships of mathematical expression format

## *3.4. Output Format*

Internally, MathEdit uses an operator-operand tree structure to support editing. Once created, an expression can be converted to and output in different formats: MathML Content, MathML Presentation, OpenMath, LaTeX, standard infix, and image formats. Each of these output formats has its own unique demands on the authoring user interface. Figure 6 shows the relationships of these formats. As mentioned before, the end user and programmer may choose and set the input method for any particular MathEdit application instance. The fact that there are multiple output formats does not affect interactive user input. The format conversion module in MathEdit can also be called directly without opening a user interface. Through this feature MathEdit can also offer a Web service for converting mathematical expression formats.

## 4. Customization

Entering and editing mathematical formulas can be important for a broad spectrum of users: students, teachers, scientists, engineers, and researchers. The broad user groups have varied requirements, contexts, and purposes for their mathematical expressions. For example, in education, requirements for mathematical notations differ for different level students: for high school students, notations for log, trigonometric functions, the Greek symbol $\pi$, and the plus-minus sign ($\pm$) as in the roots of a quadratic equation are needed while these are not necessary in elementary school. For engineers, they would rather use infix or customized shortcut key to entering expressions than operate in GUI mode. For researchers, they may want to add some new function templates which are defined by themselves.

It is impossible to design a fixed system which is appropriate for all users and all situations. Too complex a user interface can make the software a distraction rather than a help. A useful technique for MathEdit to solve these problems is to have end users tailor their editing environments and methods to match their personal work practices.

MathEdit provides various user-customizable configurations, including (i) GUI properties [27] such as the toolbar, the input palettes and other properties, (ii) input methods and output formats, (iii) editing mode [28] such as Presentation-based editing, Content-based editing or mix editing mode, (iv) initial mathematical expression for editing, (v) shortcut key for entering and editing formula, (vi) templates and corresponding MathML, and (vii) mathematical functions.

Figure 7 shows the working mechanism of customization MathEdit. MathEdit provides an easy-to-use GUI window to set all the customizable configurations. The customized result can be applied to current MathEdit window for temporary work environment, or generated MathEditCML code or JavaScript API code which can be saved in the server through server-end languages such as PHP for further use. MathEditCML is an XML-based language for MathEdit to represent and store customization configurations. Some advanced user or programmer could also edit MathEditCML code or write JavaScript code to tailor their MathEdit instance.

We have four customization levels: (1) preference setting, (2) JavaScript API codes, (3) MathEditCML, and (4) system defaults. The levels have an internal precedence. If you set the same property with distinct values at different levels a later level overrides a previous level. The level sequencing

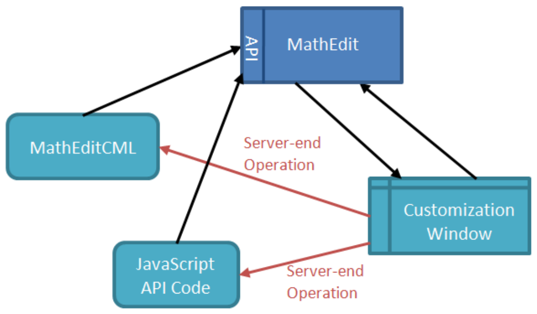


Figure 7: Customization mechanism

is from 4 to 1.

## 5. Web API

The MathEdit API Library [27] is a collection of JavaScript functions to help the Web application developer to create dynamic and engaging mathematics for Web pages. The Web API library provides a rich set of methods for creating interactive editing environments, dynamic demonstrations, and online step-by-step expository material.

The `matheditAPI.js` file provides a JavaScript wrapper class `mathedit` containing all the methods made available in the API. By using this API library, authors can create sophisticated dynamic pages with a minimum of programming. The following lists the source code of application example in Figure 4: the first line created a MathEdit instance, the 2-6 lines were used to set the edit mode, templates and GUI features, and the 7-8 lines got the infix and MathML Presentation code.

```
var matheditWin2=new mathedit("matheditWin2");
matheditWin2.set("editmode","content");
matheditWin2.set("EditorSize","286,100");
matheditWin2.set("toolbarID","tnew,tundo,tredo,tcut,tcopy,tpaste");
matheditWin2.set("templateID","s1sqrt,s1root,s1sup,s1divides,s3log");
matheditWin2.display("embed2");
var mathmlpst=matheditWind2.get("mmlpresentation");
var infixm=matheditWind2.get("infix");
```

## 6. Content-based Editing vs. Presentation-based Editing

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.

MathML supports both a *presentation encoding* and a *content encoding* for these different purposes. LaTeX is very good for the display layout of mathematical formulas whereas OpenMath, a semantic mathematical markup language, focuses on their semantics.

MathEdit addresses both display and computational aspects of mathematical expressions. The two requirements for editing mathematical expressions have duality. On the one hand, we must improve the ability for capturing semantic meaning in content-based editing and describing more layout appearances in presentation-based editing, very different requirements indeed. But our editor should provide a uniform user interface, coherent user interactions, and same output results in both of these modes.

### *6.1. Content-based Editing*

Content-based editing enables the user to enter/modify well-formed expressions that represent meaningful mathematical operations. A well-formed mathematical expression can be defined by the following description:

Given a set of *mathematical tokens* including numbers, $\pi$, $\infty$, variables/parameters (for example $x$, $y$, $\alpha$, $\beta$); and a set of *mathematical operators* (for example $+$, $-$, $*$, $/$, ^ , `sqrt`, $=$); a well-formed mathematical expression is:

- A mathematical token
- Or a mathematical operator applied to arguments which:
  - are well-formed expressions
  - are in the right number (correct aritary)

MathML Content encoding is central to MathEdit content-based editing. Content encoding can be converted to and from infix and OpenMath formats, can be converted readily to MathML Presentation for display in interactive editing, and can be converted to LaTeX and image formats with little difficulty (see Figure 6). MathEdit adopts many measures to keep the

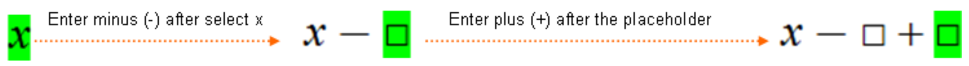

Enter minus (-) after select x
Enter plus (+) after the placeholder

Figure 8: Entering binary operator in MathEdit content-based editing

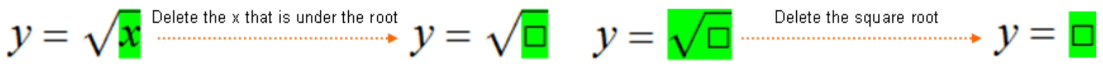


Figure 9: Deleting operation in content-based editing

expression being edited 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 8). Figure 9 also shows another example where a *placeholder* replaces a deleted sub-expression. The template menu also provides common expressions with explicit semantics for users to choose. Direct navigation of the content expression tree further ensures meaningful editing that preserves semantics.

In content-based editing the *current sub-expression* is also kept as 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 restriction. One advantage is that the user can always see and edit the infix, OpenMath, or MathML content code of the well-formed sub-expression. Another advantage is preventing the expression becoming ill-formed by the user deleting an operator. But a disadvantage is the user can't select a $+$ sign and change it directly to a $-$, say.

## *6.2. Presentation-based Editing*

Unlike content-based editing, presentation-based editing is not concerned with the semantics or well-formedness of the mathematical expression. In presentation-based editing mode, users can create arbitrary expressions that may or may not have widely accepted mathematical meaning or may not be mathematical at all. The green part of Figure 6 shows the relationship of mathematical expression formats supported in presentation-based editing.

In presentation-editing mode, MathEdit naturally use MathML Presentation encoding as its internal representation. And each edit operation, mouse click, or keyboard input, basically adjusts the DOM tree of MathML Pre-

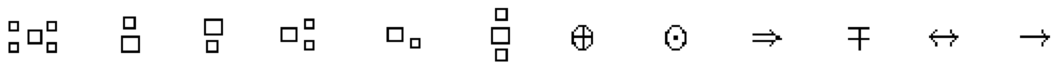


Figure 10: Some templates in presentation-based editing

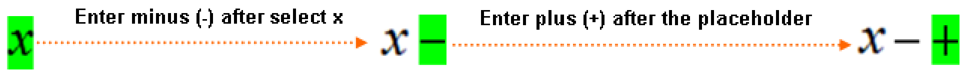

Enter minus (-) after select x
Enter plus (+) after the placeholder

Figure 11: Entering binary operator in MathEdit presentation-based editing

sentation 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. LaTeX, a format designed for typesetting mathematical formulas, can be used to interactively edit expression in the Presentation-based editing.

Presentation-based editing needs to describe the layout or appearance of an arbitrary expression. Thus, MathEdit provides additional templates for users to enter expressions with complex appearances (Figure 10). Selecting an operator in the visual navigation as the *current node* is allowed. There is also no need to use operand placeholders to preserve the well-formedness of expressions. Figure 11 shows the results with the same input operations as in Figure 8.

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 mode. Depending on the application, one of these modes can be disabled through customization.

## 7. Comparing MathEdit with Other Systems

In the past decade, many companies and research institutes have developed mathematical expression editors [8, 10, 11, 13, 14, 18, 19, 25, 26, 31]. Table 2 compared several well-known editors with MathEdit in full-visual input, character-string input, output format, editing mode (Presentation-based editing or Content-based editing).

Table 2: Compare the other editors with MathEdit

| **Item** | | **MathType** | **WebEQ** | **MathEX** | **Amaya** | **Lyx** | **TEXmacs** | **ASCIIMathML** | **MathEdit** |
|---|---|---|---|---|---|---|---|---|---|
| Full-Visual Input | Template | $\surd$ | $\surd$ | $\surd$ | $\surd$ | $\surd$ | | | $\surd$ |
| | Visual Navigation | $\surd$ | $\surd$ | $\surd$ | $\surd$ | $\surd$ | | | $\surd$ |
| | Shortcut Key | $\surd$ | $\surd$ | $\surd$ | $\surd$ | | $\surd$ | | $\surd$ |
| Character-String Input | Infix | | | | | | | $\surd$ | $\surd$ |
| | MML Content | | $\surd$ | $\surd$ | | | | | $\surd$ |
| | MML Presentation | | $\surd$ | | $\surd$ | | | $\surd$ | $\surd$ |
| | OpenMath | | | | | | | | $\surd$ |
| | LaTeX | $\surd$ | $\surd^*$ | | | | $\surd$ | $\surd$ | $\surd$ |
| Output Format | Infix | | | | | | | $\surd$ | $\surd$ |
| | MML Content | | $\surd$ | $\surd$ | $\surd$ | | | | $\surd$ |
| | MML Presentation | $\surd$ | $\surd$ | $\surd$ | $\surd$ | | | $\surd$ | $\surd$ |
| | OpenMath | | | $\surd$ | | | | | $\surd$ |
| | LaTeX | $\surd$ | $\surd^*$ | | | $\surd$ | $\surd$ | $\surd$ | $\surd$ |
| | Picture | $\surd$ | $\surd$ | | | | | $\surd$ | $\surd$ |
| Content-based Editing | | | | $\surd$ | | | | | $\surd$ |
| Presentation-based Editing | | $\surd$ | $\surd$ | | $\surd$ | $\surd$ | $\surd$ | $\surd$ | $\surd$ |
| Web-based | | | $\surd$ | $\surd$ | | | | $\surd$ | $\surd$ |
| Web API | | | $\surd$ | $\surd$ | | | | | $\surd$ |

Let us first focus on input style and output format. Most of the editors can support visual direct-manipulation: they provide well-defined templates, visual navigation by mouse, arrow keys, and shortcut keys. But MathEdit, MathEX [20], and WebEQ [26] support more flexible customizable templates. Although many prefer to use templates and the mouse when first learning an application, in the long run it is often more convenient to use keyboard shortcuts for common operations. All the editors use either MathML Presentation, or LaTeX, or both as their main mathematical representation format. For character-string input, LaTeX is the most popular input format which is supported in MathEdit, MathType [6] (MathType Version 6 begins to support LaTeX input), TeXmacs [23], ASCIIMathML [2] and WebEQ (WebEQ use WebTeX which syntax and commands are similar to the mathematics mode part of LaTeX). The table also shows us supporting for directly editing MathML Presentation is also an important feature for the modern mathematical expression editor. Only MathEdit and MathEX can support OpenMath which is a semantic markup language. MathEdit is only one which supports combination of infix and direct-manipulation input editing method.

For the editing mode, the other editors usually aim either to capture the meaning or to describe the visual appearance. The Amaya [1], LyX [12], TeXmacs and MathType using MathML Presentation, LaTeX, or native formats to store expressions are suitable for describing the expression appearance. Most Computer Algebra Systems, such as Maple [4] and Maxima [7], and a few independent editors, such as MathEX, use infix and MathML Content to capture the meaning of expressions. MathEdit is different which satisfies both the need for visual display and the need for expression processing.

For the three Web-based visual editors, MathEX, WebEQ, and MathEdit, the first two are Java Applets whereas MathEdit is a JavaScript object that can interact with its hosting Web page seamlessly. Because MathEdit is open source, application programmers can also create additional ad hoc APIs to MathEdit for their own purposes.

## 8. 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, 60773108, and 60473095, and Ministry of Science and Technology of China under Grant 2005DKA64001. 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] ASCIIMathML Homepage http://www1.chapman.edu/ jipsen/mathml/asciimath.html

[3] Document Object Model http://www.w3.org/DOM/

[4] Document of Maple http://www.maplesoft.com/Products/Maple/

[5] Document of Mathematica http://reference.wolfram.com/mathematica

[6] Document of MathType http://www.dessci.com/en/products/mathtype

[7] Document of Maxima http://maxima.sourceforge.net/

[8] 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.

[9] LaTeX documentation http://www.latex-project.org/guides/

[10] Luca Padovani and Riccardo Solmi, *An Investigation on the Dynamics of Direct-Manipulation Editors for Mathematics*, MKM 2004, LNCS 3119, pp. 302-316, 2004.

[11] Jean-F. Nicaud, *Natural Editing of Algebraic Expressions*, Proceeding of Mathematical User-Interfaces Workshop 2007, Schloss Hagenberg, Linz, Austria, June 2007.

[12] LyX http://www.lyx.org/

[13] M. Pollanen, T. Wisniewski, and X. Yu, *Xpress: A Novice Interface for the Real-Time Communication of Mathematical Expressions*, Proceeding of Mathematical User-Interfaces Workshop 2007, Schloss Hagenberg, Linz, Austria, June 2007.

[14] Mathmled http://www.newmexico.mackichan.com/MathML/mathmled.htm

[15] MathPlayer http://www.dessci.com/en/products/mathplayer/

[16] Murray Sargent III, *Unicode Nearly Plain-Text Encoding of Mathematics*, Unicode Technical Note, http://www.unicode.org/notes/

[17] 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 Southeast Conference, 2005.

[18] 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 August.

[19] Samuel S. Dooley, *Editing Mathematical Content and Presentation Markup in Interactive Mathematical Documents*, Proceedings of ISSAC, 2002.

[20] 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 2007.

[21] Standard ECMA-262 http://www.ecma-international.org/

[22] 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.

[23] TeXmacs http://www.texmacs.org/

[24] The OpenMath Standard 2.0 Draft http://www.openmath.org/

[25] The W3C MathML software list http://www.w3.org/Math/Software/

[26] WebEQ Documentation http://www.dessci.com/en/products/webeq

[27] 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.

[28] Wei Su, Paul S. Wang, Lian Li, *Entering and Editing Mathematical Expressions on the Web*, Proceeding of Mathematical User-Interfaces Workshop 2008, University of Birmingham, UK, July 2008.

[29] 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.

[30] W3C Math http://www.w3.org/Math

[31] 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.