

# An On-line MathML Editing Tool for Web Applications

Su Wei

Lanzhou University, Lanzhou,  
Gansu, 730000, China  
suwei@lzu.edu.cn

Paul S.Wang

Kent State University, Kent,  
Ohio 44242-0001, USA  
pwang@cs.kent.edu

Li Lian

Lanzhou University, Lanzhou,  
Gansu, 730000, China  
lil@lzu.edu.cn

## Abstract

MathEdit [23] is a browser-based tool implemented in JavaScript that provides a convenient and intuitive graphical user interface for creating and editing mathematical expressions as well as a well-defined API for interoperating with its containing Web page. MathEdit can produce MathML Content and Presentation Markup as well as infix expressions for Web applications. The results returned can be displayed in a Web page or used in further computations. Here, we describe the methods of expression editing, visual navigation, and the MathEdit API. We also present the actual use of MathEdit in such interactive web applications as curve plotting and Math-capable BBS.

## 1. Introduction

The ability for users to interactively create and manipulate mathematical expressions is very important for systems such as Science Communication Environment, Computer Algebra Systems, and Web-based Mathematics Education [8, 14, 15, 25, 26]. MathEdit is a browser-based mathematical expression editor developed jointly by Lanzhou university/China and Kent University/USA [2, 23, 27]. Running in a Web browser, MathEdit allows users to create and edit mathematical expressions with a convenient and intuitive graphical user interface and programs to access it with a well-defined API. With MathEdit, users can easily enter mathematical expressions online as answers to questions in mathematics lesson pages [5, 22] for example. Web page authors can also embed MathEdit in their interactive web page to create mathematical expressions. Figure 1 shows the main MathEdit user interface (boxes with black lettering are part of the illustration). The extent of the templates menu is easily customizable to support different levels and areas of mathematics. MathEdit is implemented in standard JavaScript [17] and DOM [3]. MathEdit runs within any standard Web browser that supports JavaScript and DOM. Figure 2 shows the end-user view of the MathEdit architecture. User actions, mouse clicks and keyboard input, are treated as commands. Commands invoke JavaScript functions in the command processing module. The main command processing modules include

expression editing and visual sub-expression navigation. Each edit operation basically adjusts the DOM tree of MathML content markup [2, 9, 20] kept internally for the mathematical expression being constructed or edited.

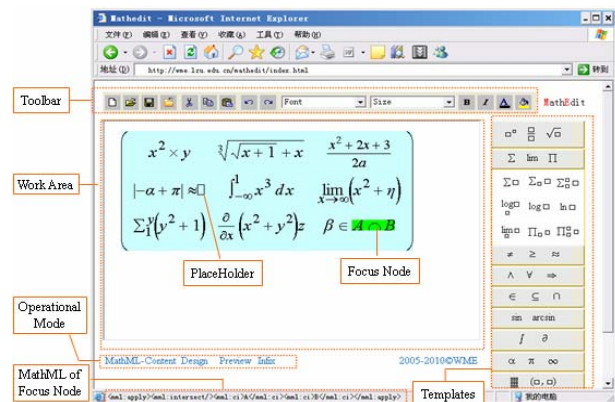


Figure 1: MathEdit Authoring Environment

MathEdit can serialize the content tree to MathML Content, Presentation, and Mixed Encodings as well as to an infix string. For reading an existing expression, we use the Mozilla XML Dom parser under the Mozilla family browsers [10, 11] and the MSXML parser under Microsoft IE to conveniently process the MathML input. The *Mathematical Expression Toolbar* is customizable and extensible via adding and deleting templates. New special characters, such as  $\infty$ , Greek Characters, etc, can also be introduced by the user through a combination of keyboard and mouse operations.

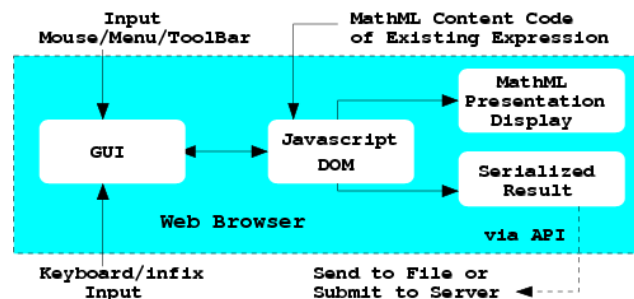


Figure 2: MathEdit Architecture

We begin by describing the editing and user interaction parts of MathEdit. Then we'll show how this tool can be used, through its API, to add mathematical formula capabilities in other Web applications.

## 2. 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 supports several forms of user input, infix, MathML and template. Infix is convenient for users who are familiar with the notation. For others, the common approach is to use graphical templates [1, 4, 7, 16, 19, 21] that can be selected to enter particular kinds of mathematical expressions such as fractions, square roots, powers, and so on.

### 2.1 Templates

Graphical icons represent mathematical special characters and expression patterns. Each item on the *template panel* expands to reveal a group of related templates (Figure 3). Combining infix input and template selection (via the mouse or keyboard shortcut) makes it easy and flexible for users to create expressions. After inserting a template, the internal DOM tree and the display are updated with required expressions indicated by *placeholders*. The user can select each placeholder to enter the desired expression.

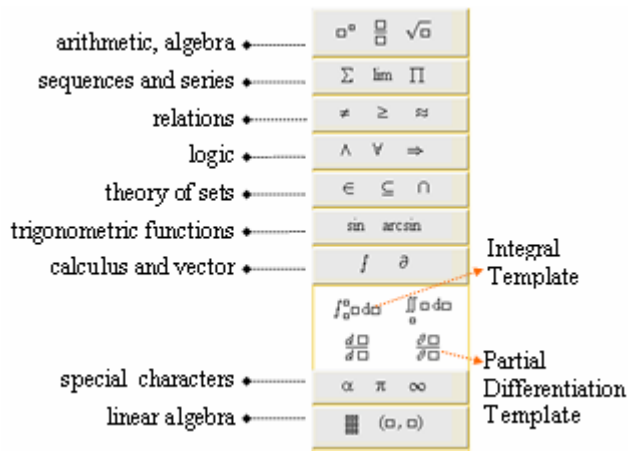


Figure 3: the Template Panel

### 2.2 Entering Expressions

Expression editing is done by selecting an insertion point or a sub-expression to replace and then entering an expression. The command processor module adjusts the target node on the MathML content tree through DOM.

Token input is the most basic operation. Three kinds of input tokens, digit (0 through 9 and the decimal point), letter (alphabetical character, mathematical symbols and Greek characters) and sub-expression (including operator), can be entered by keyboard or mouse. The command processor analyzes the input token incrementally as it is received and displays it immediately. Table 1 shows how the treatment of a new token (row headings) is dependent on the previous token (column headings).

MathEdit makes it easy to correct typing errors. The `Backspace` key deletes the last character typed that is still in the input buffer. The `Backspace` key has no effect if the input buffer is empty. The `Delete` key erases the current expression and replaces it with a placeholder if necessary to maintain the integrity of the MathML content code.

### 2.3 Editing Modes

MathEdit supports four operational modes: *Visual Edit mode*, *MathML mode*, *Infix mode*, and *Preview mode*. Generally, the *Visual Edit mode* is for users to create and edit mathematical expressions interactively. In *MathML mode*, a user can view, enter, and edit MathML source code directly. Copy and paste of MathML code is supported. In *Infix mode* a user can enter expressions in infix notation or extract the infix representation of the expression being edited. In *Preview mode*, the expression is displayed in its final form and no editing is allowed. Previewing is usually the last step before a user is done creating or editing an expression.

In *Visual Edit mode*, the MathML Content code of the currently selected expression is always displayed in the MathEdit status bar. It shows, at a glance, the nesting of the MathML structure at the cursor location, and gives users an easy way to select sub-expressions containing the cursor. The current sub-expression can be edited in a pop-up window and the edited code replaces the old code when the pop-up window is closed. Alternatively, a user can choose to edit the infix string for that sub-expression instead.

Table 1: Inputting Operation Methods in MathEdit

New Token Previous Token	Letter	Number	Templates or Operator
Letter	Append the new letter. E.g. $a \leftrightarrow b \Rightarrow ab$	Append the new number. E.g. $a \leftrightarrow 2 \Rightarrow a2$	Add new sub-expression and the focus node will be the part of sub-expression. E.g. $\pi \leftrightarrow \frac{\square}{\square} \Rightarrow \frac{\pi}{\square}$
Number	The number multiplies the letter. E.g. $3 \leftrightarrow b \Rightarrow 3b$	Append the new number. E.g. $3 \leftrightarrow 2 \Rightarrow 32$	Add new sub-expression and the focus node will be the part of sub-expression. E.g. $3 \leftrightarrow \sqrt[\square]{\square} \Rightarrow \sqrt[\square]{3}$
Sub-expression	Replace the sub-expression with new letter. E.g. $\lim_{x \rightarrow \infty} (x^2 + 1) \leftrightarrow y \Rightarrow \lim_{x \rightarrow \infty} y$	Replace the sub-expression with new number. E.g. $\frac{d}{dx} y \leftrightarrow 3 \Rightarrow \frac{d}{dx} 3$	Add new sub-expression and the focus sub-expression will be the part of new sub-expression. E.g. $(\beta \in A \leftrightarrow (\square \cup \square)) \Rightarrow (\beta \in A \cup \square)$
Placeholder	Replace the placeholder with the letter. E.g. $\square \leftrightarrow \theta \Rightarrow \theta$	Replace the placeholder with the number. E.g. $\square \leftrightarrow 5 \Rightarrow 5$	Replace the placeholder with the template. E.g. $\square \leftrightarrow \sum_{\square} \square \Rightarrow \sum_{\square} \square$

### 3. Visual Navigation

Editing operations, such as insert, edit, delete, covert, and replace, are relative to the *current node* (a node on the MathML Content tree) and the *cursor position* (blinking vertical line). The current node expression (the current expression) is highlighted visually and visual navigation refers to moving the current expression and the cursor 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*.

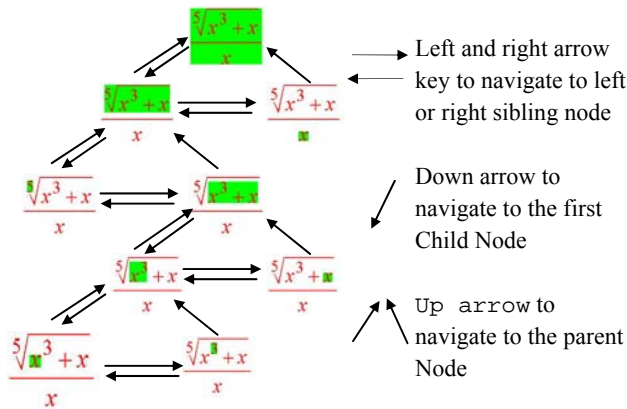


Figure 4: Basic Navigation

#### 3.1 Basic Navigation and Traversal Navigation

The four arrow keys are used for Basic Navigation. They move the current sub-expression up to the parent node, down to the first child, or left/right to sibling nodes (Figure 4).

Because Basic 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). The PageUp key, on the other hand, provides the inverse-orient traversal. Figure 5 shows the flow chart for the algorithm used for PageUp key navigation.

To make navigation more intuitive, MathEdit also allows the user to use the mouse to move the current node.

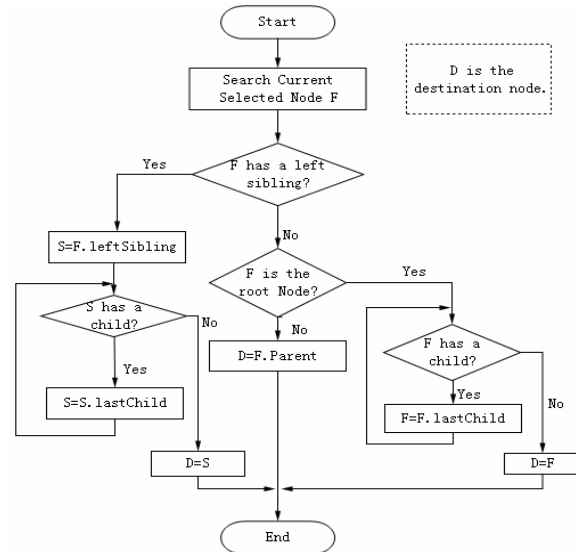


Figure 5: Traversal Navigation

#### 3.2 Cursor Movement

In MathEdit the location of the cursor is indicated by a blinking vertical line. The cursor indicates the current insertion point and may be positioned either to the left or right of the current expression. The cursor can be moved with the mouse or the arrow keys. In general, the editor will move the cursor to the next valid insertion location in the direction of the arrow key or nearest to a mouse click. To implement this feature, MathEdit must negotiate the MathML Content Expression Tree. Moving the cursor to the right on the screen can involve non-trivial traversals on the DOM tree. The following DOM code shows the cursor display technique:

```
var domfrag=
designDoc.createDocumentFragment();
var tag=
designDoc.createElementNS(mmlns, "mi");
tag.setAttribute("id", "cursor");
tag.setAttribute("mathcolor", "#0f0");
var tnode=designDoc.createTextNode("|");
tag.appendChild(tnode);
domfrag.appendChild(tag);
```

To make the cursor display work, an artificial `<mathphantom>` element is sometimes inserted to the DOM tree which will either be replaced by a new sub-

expression entered by the user or removed when the cursor is moved elsewhere.

#### 4. MathEdit API and Applying MathEdit in Interactive Web.

The MathEdit API, supported by a set of JavaScript functions, allows applications to create an editor instance, sets any initial expression in it, configure its template palette and other GUI features, and set/retrieve the MathML code it contains.

##### 4.1 Two Methods of Applying MathEdit

Table 2 lists most API functions of MathEdit API Library. There are two methods of applying MathEdit into interactive web: pop-up window and in-page frame.

In pop-up window, user can activate MathEdit by `newMath()` or `editMathById()`. A pop-up window with MathEdit will open and user can edit their desired math expressions by MathEdit. When user clicks Return button, the object math code (MathML or Infix) will transfer to the main program.

In in-page frame, the MathEdit can be embedded in interactive web to edit math expression. User can customize the templates and toolbars through MathEdit API function `showTemplate(templateID)` and `showToolbar(toolID)`.

Table 2: The MathEdit API Library

Method	Description
<code>newMath( id, parentId )</code>	Create a new MathML expression.
<code>setIframe(iframeID,matheditURL)</code>	Set an Iframe with an ID attribute to Embed MathEdit in your Web.
<code>editMathById( id )</code>	Edit an existing Math Expression by ID.
<code>showTemplate(templateID)</code>	Display the mathematical template by template ID.
<code>showToolbar(toolID)</code>	Display the tool button by ID.
<code>setEditorSize(width,height)</code>	Set the size of editor area.
<code>loadContentMath(filename )</code>	Load Content MathML from a file
<code>loadPresentationMath(filename )</code>	Load Presentation MathML from a file
<code>loadInfixMath(filename )</code>	Load Infix math expression from a file
<code>editContentMath(String)</code>	Load Content MathML from a string
<code>editPresentationMath(String)</code>	Load Presentation MathML from a string
<code>editInfixMath(String)</code>	Load Infix math expression from a string
<code>getContentMathML ()</code>	Return Content MathML to host procedure
<code>getPresentationMathML()</code>	Return Presentation MathML to host procedure
<code>getInfixMath ()</code>	Return infix Math expression to host procedure

##### 4.2 Plotting User-Entered Equations

To demonstrate the use of the MathEdit API, let's look at a sample WME page (Figure 6) for graphing an equation entered by the user. The form has two entries. The user first clicks on an icon to pop up the MathEdit tool in order to create a desired formula

interactively. When done, the user clicks the return folder icon on the menu bar to close the MathEdit window. The resulting MathML is retrieved via the MathEdit API and displayed in the step-1 entry via DOM operations.

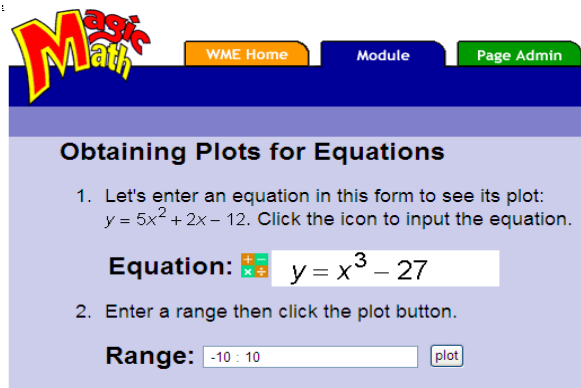


Figure 6: A Sample WME Page of Applying MathEdit

After filling in the plotting range, a click on the `plot` button triggers JavaScript code, in the enclosing Web page, which obtains the string representation of the formula created via the MathEdit API. The formula string then becomes part of a POST request to a curve plotting program which displays the result (Figure 7).

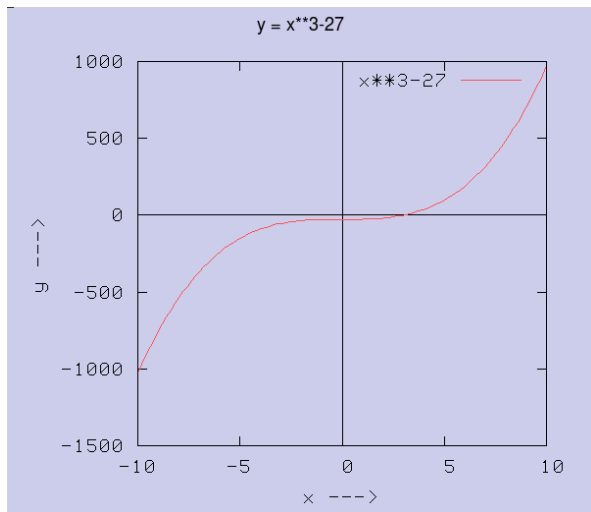
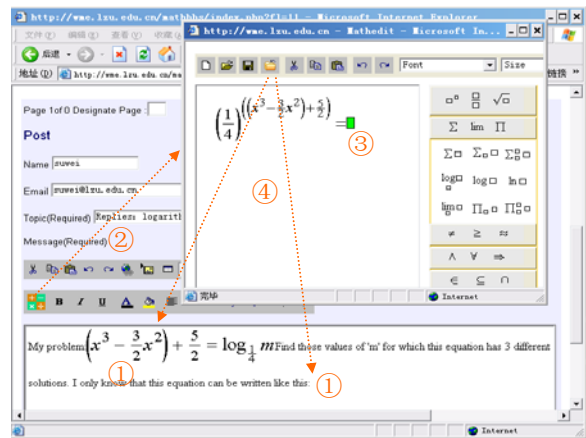


Figure 7: Plotting User-Entered Equations

### 4.3 Math-enabled BBS

Bulletin board systems (BBS) are on-Web forums for posting discussion topics, questions, responses, and comments. A useful tool for WME [13] is MathBoard, a math-enabled BBS for mathematics educators and students that can support online discussions. For a mathematical bulletin board (MathBoard), support for mathematics expressions and diagramming must be provided. MathBoard implements online mathematical expressions editing powered by MathEdit and online mathematical graphic and geometry editing powered by GeoSVG [24].



- ① Set insert focus or select existed expression
- ② Call MathEdit
- ③ Edit expression
- ④ Return MathML to insert point or replace existed expression

Figure 8: Using MathEdit in MathBoard

Figure 8 shows the MathBoard GUI and the main steps to post a new or update an existing mathematical expression. To post a new formula, a user simply clicks the MathEdit icon to popup the tool. The expression created is then inserted at the cursor position. If the user selects an existing expression (highlighting it with the mouse) before clicking the icon, then the MathEdit popup window will display that expression for editing. The editing result will then replace the target expression. Mathematical expressions have IDs unique in their page to make DOM operations on them easy. The following shows the essential JavaScript code for MathBoard to use MathEdit.

Table 3: The Essential Code in MathBoard To Apply MathEdit

```

if(editor.document.selection)
{
    var sele=editor.document.selection;
    var sel =sele.createRange();
}
if(sele.type=="Control")
{
    for (i = 0; i <sel.length; i++)
    {
        mathtID=sel(i).id;
    }
    mathedit.editMathById(mathtID);
}else
{
    var mathtID="math"+(mathID++);
    mathedit.newMath(mathtID, "cursor");
}

```

## 5. Future Work

Work on MathEdit is continuing. MathEdit now has an IE version that works with the MathPlayer plugin and a Mozilla version that works with native MathML support in Firefox, Netscape Navigator, and Mozilla. There are some differences between the two versions, even at the user level. We are looking to integrate the two versions into a cross-browser tool. We also want to put MathEdit to more extensive use in different scientific, technical and educational systems, to collect feedback, and to make improvements. User feedback will be important to fine tune the GUI and to refine the MathEdit API. Support for additional mathematical encoding formats such as OpenMath [18], MP [12], and LaTeX [6] may also be provided.

## 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 and Dr. Xun Lai of Kent State University for making available his GeoSVG for use in the MathBoard.

The material reported here is based upon work supported in part by the National Natural Science Foundation of China under Grant No. 90612016, 60473095 and the National Science Foundation of USA under Grant CCR-0201772. 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.

## Reference

[1] Amaya Homepage  
<http://www.w3.org/Amaya/>.

[2] Cai Chuan, Su Wei, Li Guanyu, Zhao Yanjuan, and Li Lian, *The research of MathML editing and displaying technology*, Proceeding of China National Theoretical Computer Science Conference, China, 2006.

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

[4] Document of MathType  
<http://www.dessci.com/en/products/mathtype>.

[5] Guanyu Li, Lian Li, Wei Su, and Yanjuan Zhao, *Design and Implementation of MAML*, Proceedings of ATCM, Hong Kong, China, 2006.

[6] LaTeX documentation  
<http://www.latex-project.org/guides/>.

[7] LI Junguo, ZHANG Lipang, *A XML Based Presentation Technique for Mathematical Expressions on Web Page*, Acta Scientiarum Naturalium Universitatis Pekinensis, Vol. 39, No. 5 (Sept, 2003).

[8] Luca Padovani, *Interactive Editing of MathML Markup Using TEX Syntax*, TEX, XML, and Digital Typography 2004.

[9] MathML Conference 2002  
<http://www.mathmlconference.com/2002/>.

[10] Mathmled  
<http://www.newmexico.mackichan.com/MathML/mathmled.htm>.

[11] Mozilla MathML Project  
<http://www.mozilla.org/projects/mathml/>.

[12] MP: The Multi Protocol  
<http://www.symbolicnet.org/areas/protocols/mp.html>.

[13] 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.

[14] Samuel S. Dooley, *coordinating Mathematical content and presentation markup in interactive mathematical documents*. Proceeding of the 1998 International Symposium on Symbolic and Algebraic Computation, ACM SIGSAM, ACM Press.

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

[16] Samuel S. Dooley, *Users Guide for the Integre MathML Equation Editor*  
<http://www.integretechpub.com/mathex/>.

[17] Standard ECMA-262  
<http://www.ecma-international.org/>.

[18] The OpenMath Standard 2.0 Draft  
<http://www.openmath.org/>.

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

- [20] W3C Math <http://www.w3.org/Math/>.
- [21] WebEQ Documentation  
<http://www.dessci.com/en/products/webeq>.
- [22] Wei Su, Guanyu Li, Yanjuan Zhao, Lian Li, *Initial Design of Mathematics Assessment Grid*, Presentation at Symbolic Computation in Education'2006.
- [23] Wei Su, Paul Wang, Lian Li, Guanyu Li, and Yanjuan Zhao, *MathEdit, A Browser-based Visual Mathematics Expression Editor*, Proceedings of ATCM 2006, Hong Kong, China, 2006.
- [24] Xun Lai, Paul Wang, *An SVG Based Tool for Plane Geometry and Mathematics Education*, IEEE SoutheastCon, 2005.
- [25] Yasutomo Nakayama, *Mathematical Formula Editor for CAI*, CHI'89 PROCEEDINGS.
- [26] 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.
- [27] Zhao Yan-juan, Li Lian, Su Wei, and Li Guan-yu, *Implementation of MathML-Based Web Mathematical Expressions Editor*, Computer Engineering (accepted).