Building A Browser-Based Visual Editor for Mathematical Expressions
MathEdit Collaboration

- Su Wei and others at Lanzhou University, PRC
- Paul Wang at ICM/Kent, USA
What is MathEdit

- A self-contained visual editor for Math that is easily interfaced to WME and other Web applications.
- MathEdit is implemented in standard JavaScript uses DOM to represent the mathematical expression being created/edited.
- MathEdit can produce MathML code—content, presentation, and composite codes.
- Its GUI offers visual navigation of sub-expressions and an expression template palette for expression input.
Previous Work

- Early math editors use proprietary formats making interoperability with other programs a problem.
- Formulas displayed as images are not re-usable.
- Recent MathML-based editors (MathPlayer, IBM TechExplorer) still use proprietary mathematics rendering engines.
MathEdit Features

• MathEdit makes use of browser support for MathML.
• MathEdit is open and coded entirely in standard JavaScript.
• MathEdit runs inside any MathML-capable Web browsers and can be included in any Web page that requires mathematical expression input.
• MathEdit allows both infix and template-based input.
• MathEdit has a well-defined customization and extension mechanism.
\[-\frac{b \pm \sqrt{b^2 - 4ac}}{2a}\]
MathEdit Functionalities

- Create a new or edit an existing mathematical expression interactively with a convenient GUI
- Direct editing of MathML code
- Set mathematical expression format and style
- Customize toolbar, palette and expression template
- Import/Export MathML
- Capture and retrieve the MathML markup from other applications or webpage.
- Open an existing MathML file stored in the local file system or at the originating Web server
- Save MathML in a local or remote file
• Return the result mathematical encoding (content, presentation, and composite) to the parent window
A Usage Scenario

1. The user accesses a Web page using a suitable browser, say Firefox.

2. The Web page, together with MathEdit (JavaScript code), is delivered to the user’s browser which displays it.

3. The user now fills out a form on this page which contains an entry for a user-created mathematical expression.

4. The user clicks on an ”Enter Expression” button which invokes MathEdit in a pop-up window for the user to enter the desired expression.

5. The user clicks a ”Done” button to close the pop-up window causing the finished expression to be returned to the form and displayed together with the rest of the form.
6. The user submits the form after filling it all out. The form data, including the mathematical expression, coded in composite MathML, is posted to a specified server.
**Design and Implementation**

- Infix input and input via Visual expression template palettes.
- Sub-expression navigation a direction reflection of DOM tree traversal.
- User commands (input) results in a DOM tree fragment that is placed on the DOM tree which represents the entire expression being edited.
- Infix string parsing borrows from Xiao Zou’s parser.
- Rendering of math expressions is by browser/plugin support of MathML.
Need for Extensibility and Customization

- It is important to carefully balance ease of use, power and functionality, and the needs of different user groups.
- MathEdit has a broad spectrum of potential users: students, teachers, scientists, engineers, and researchers.
- In secondary education, students and teachers need to be able to create mathematical content quickly and easily through an intuitive and natural user interface.
- Requirements for mathematical notations differ for different users.
Support for Extensibility and Customization

- The `config.xml` file contains customization info. At initialization time, MathEdit will load and parse the server-side `config.xml` file.

- Application developers can directly edit the configuration file to customize MathEdit.

- For certain end uses, we also provide a dialog window to change the configuration information. Customized results are saved back to the server.

- Customizable configurations include GUI properties, the toolbar, the input palettes and other properties.

- GUI properties include font size, color, window background color, window size, highlight color, etc.
• The toolbar contains the file menu, the edit menu, and the format menu. The MathEdit palette displays a set of icons for the available input templates.

• The main palette may contain template icons as well as sub-palettes that pull down with a mouse click.
A set of JavaScript functions allowing applications to:

- Create an editor instance (associated with a pop-up window or in-page frame)
- Set any initial expression in it
- Configure template palette and other GUI features
- Set/retrieve the MathML code it contains.
Further Work

- Test and refine the user input and navigation model.
- Add user-level *Copy and Paste*.
- Support additional output formats such as \textsc{LaTeX}.
- Make Cross-browser implementation.