WME MathEdit Design & Requirements Analysis

K. Cem Karadeniz
# Table of Contents

I. Introduction........................................................................................................................................3

1.a. Purpose of This Document..................................................................................................................3
1.b. Scope of This Document....................................................................................................................3
1.c. Document Conventions....................................................................................................................3
1.d. Intended Audience.............................................................................................................................3

II. Application Overview..........................................................................................................................3

2.a. Objectives........................................................................................................................................3
2.b. User Roles and Responsibilities.........................................................................................................5
2.c. Interactions with Other Systems..........................................................................................................6
2.d. Constraints, Open Issues, and Questions..........................................................................................6

III. Functional Requirements.....................................................................................................................8

3.a. Statement of Functionality..................................................................................................................8
3.b. Interface Specification.........................................................................................................................9
3.c. Security...........................................................................................................................................11
3.d. Auditing...........................................................................................................................................12
3.e. Administration/Customization of the Application.............................................................................12
3.f. Scope............................................................................................................................................12
3.g. Performance..................................................................................................................................13
3.h. Usability.......................................................................................................................................13

IV. Competitive Analysis........................................................................................................................14

4.a. Comparison of Software Solution with Existing Approaches............................................................14
I. Introduction

1.a. Purpose of This Document

This document details the design and the functionality required for the WME MathEdit.

1.b. Scope of This Document

This document is intended as a set of requirements and technical information.

1.c. Document Conventions

The words “application”, “software program”, and “editor” are used interchangeably to refer to the WME MathEdit, and the word “expression” refers to any kind of mathematical notation or mathematical expression.

1.d. Intended Audience

The primary audience is the WME research team. This includes programmers, teachers, students and testers of the team. Ultimately, all users of the WME software and who are interested in web-based education.

II. Application Overview

2.a. Objectives

Why this application is needed? Modern GUI-based mathematical authorizing tools usually have options to save documents as web pages. What these tools basically do is to save your mathematical content as images (see Fig. 1) and text as HTML. This works good if you want to publish static content such as an online paper with mathematical formulas which never need to change or mathematical thesis research, in other words if there is no dynamic content and no need for semantics. But advanced mathematical web sites should publish content dynamically from back-end databases and should allow users
WME MathEdit

Design and Requirement Analysis

WME MathEdit is intended to work with the WME system which uses W3C's MathML to deploy mathematics on web pages. MathML is designed for being human legible (though it is very verbose), and simple for software to generate and process. Since the code of MathML can easily become a complex puzzle to describe mathematics in most of the cases, it is quite fair to expect from our computers to generate MathML code for us. Therefore, there should be some intermediate tool (software) which accepts input combined with visual items as if we are writing mathematics on a piece of paper and outputs the MathML representation for us. Then the same tool could send the output that it creates to another program or directly to a web page as in the case of WME. As a matter of fact, we cannot expect teachers and students who are using WME to learn how to code MathML to create their mathematical expressions while preparing questions or answering them.

Figure 1: Saving as static mathematical content

to create/submit/edit their own content to share with users. This is crucial for a web and mathematics-oriented project such as WME which has a sophisticated assessment system. After all, visual representation of mathematics with its associated semantics is a basic need for such a project.
WME MathEdit is a software program that is going to fill this gap between the user and the computer. It is going to help the user to encode mathematical material suitable for web viewing.

2.b. User Roles and Responsibilities

The following list describes the users of the application:

- Teachers

  The teachers who use the WME systems to teach mathematics constitute the primary user set. They employ the application while creating, customizing, and editing topic lesson pages (TLPs), preparing questions/answers for the Distributed Mathematics Assessment Database (DMAD), or similar circumstances where they need to print dynamic mathematical content.

- Students

  The students who are being taught with the WME systems use the application when answering questions as part of the DMAD system, creating and saving their own mathematical expressions, or similar circumstances where they need to print dynamic mathematical content.

- WME system administrators/developers

  The researchers of the WME system use the application for creating dynamic mathematical content as part of their individual projects and for administering the WME web pages.
2.c. Interactions with Other Systems

Interaction of WME MathEdit with other systems is through MathML which is created for being a way to allow the passing of information intended for specific renderers and applications, encoding both mathematical notation and mathematical meaning (semantic). WME MathEdit creates MathML code, this makes the application interact with any system which expects MathML code as input to manipulate. In the case of WME this interaction occurs in two ways (see Fig. 2).

Figure 2: WME MathEdit interactions with the WME systems

2.d. Constraints, Open Issues, and Questions

Considering the notion of the WME project WME MathEdit should be a web-based editor therefore the application must run online. This brings some constraints as well as some advantages. The major advantage of being a web-based application is the flexibility of platforms required to run the application. In most of the cases a browser with the appropriate plug-in is enough to use the software, plus the user does not need to go through an installation process to run the application.

One of the drawbacks of being an online application is the bandwidth requirement. The software should provide an acceptable response time to the user.
commands. This deals highly with the functionality and the selected tool(s) to implement the application. These points should be clarified during the requirements elicitation process.

At the first look there seems to be two alternatives for the selection of the tools to implement the application: Macromedia Flash and W3C's Scalable Vector Graphics (SVG) technology. The first option comes with a great plug-in support and an accepted market platform. Although a detailed research about Macromedia Flash, it has been shown that it cannot meet a crucial requirement of the application which is ability to edit an existing mathematical expression which is already included in a web page. This deficiency comes from the nature of the tool, it does not have support for MathML.

On the other hand the second alternative (SVG) meets all the requirements including the editing capabilities and JavaScript communication (not all platforms). This free technology is still under a slow development process which makes it desirable for academic projects. The major drawback of this tool is its poor plug-in support. At this time only MS Internet Explorer can render SVG with a decent plug-in under MS Windows environment. It seems that the future of SVG is still unknown. WME MathEdit is first designed to be implemented under Macromedia Flash with its powerful scripting language ActionScript 2.0 and JavaScript. But considering the requirement of the editing capabilities of the application this tool was found to be inadequate for this purpose. At this time the primary concern about the application is the ability to capture an existing mathematical expression and draw it on the application's typing area (see Fig. 4 on page 10).
III. Functional Requirements

3.a. Statement of Functionality

This section lists the functional requirements in ranked order. Mathematical expressions mentioned below are the code of MathML, therefore they are dynamic. WME MathEdit is expected to:

1. create a new mathematical expression using its user-friendly interface.

2. capture and draw an existing mathematical expression in an appropriate way to edit/customize it.

3. edit an already created and deployed mathematical expression, this has to be preceded by the second list item.

4. save the newly created or customized expression's MathML code in an external file as text.

5. show the MathML code created and/or running behind.

6. allow the user to manually edit the MathML code of the expression on the run.

7. send the newly created or customized expression back to the parent application which is MathML-capable web pages in the WME structure. The page then must be reloaded in order to render the obtained MathML code.

These functional statements are based on the user input and the editor-generated associated MathML code (see Fig. 3).
3.b. Interface Specification

The interface of the application should be easy and user-friendly enough to be able to used by the teachers and the students who have never used the WME system. The interface together with the functionality should not require any extra time to learn how to create simple mathematical expressions. The current interface of the application can be seen below (Fig. 4). This interface design is formed with consideration for Macromedia Flash as the development tool. Therefore, it is very tentative due to the development environment. The same design cannot be achieved using the SVG with a viable running time of the application considering the current plug-in support. Also the lack of a competitive authoring environment (or a Rapid Application Development environment – RAD) for SVG might cause dramatic changes in the interface.
The interface of the application is designed to meet the following Graphical User Interface (GUI) operations:

1. Accept textual user input in the form of MathML code
2. Accept drag-drop user input which allows user to select any MathML rendered expression to be edited (tentative)
3. Display (draw) the expression
4. Select and drag-drop the expression on the typing area for output
These GUI operations can be visualized as in the figure 5 below.

![Diagram of GUI operations]

*Figure 5: GUI operations*

### 3.c. Security

As stated above (section 2.b), the WME project is used/authored mainly by three groups of people: teachers and students who utilize the WME, and the WME developers who maintain the software. The application will be working under the general security organization of the WME system; whoever is eligible to use WME should be authorized to use the editor. There is no special requirement for the editor in terms of security (like a login system) other than the WME systems has already provided. As a matter of fact, it is not difficult to enable the application to work outside of the WME scope without any restricted user access.
3.d. Auditing

The meaning of auditing in this context is being able to tell who made changes, when they were made and what they were. However, the application is being developed by one author therefore there is no auditing features required for the development process.

3.e. Administration/Customization of the Application

Administration and customization of the software program is maintained by the WME developers. When required (rarely), the qualified WME teachers might also customize the editor according to their specific needs for teaching.

3.f. Scope

At the first phase, the application is planned to cover the needs for the 6th and 7th grades mathematical curriculum. This includes basic mathematical operations (+, -, x, ÷), square root, second and third powers, fraction, simple functions (i.e. sin, cos, tan) and, selected mathematical symbols according to their needs. After having provided the basic functionality, as a second phase, the application should incorporate more mathematical content. This includes set theory symbols and operators, trigonometry, calculus (i.e. derivation, integration functions and notations), special functions (i.e. ceiling, floor, absolute value) and so on. The scope of the application is strictly related to the development environment since it will affect the interface and code dramatically. The response time of the application should be kept minimum for an efficient utilization. The need for a well designed and organized interface increases as the content that the application covers increases. The current interface design is powerful enough to cover a good number of mathematical content (see Fig. 6).
3.g. Performance

As mentioned earlier, the performance is one of the most critical concerns for the application. The concern is even more important when thinking of the SVG technology as the development tool. Because of the cumbersome plug-in implementation SVG applications become slow-running programs, increasing response time and decreasing efficiency. Considering the fully implemented version of the editor this might create more problems at the second phase.

3.h. Usability

There is no specific usability requirement for the application other than that of the WME
system software. Speed of navigation of the user interface is a usability criterion. Using SVG, being the only alternative for now, the speed of navigation (i.e. response time) of the interface will be an important goal to achieve during the development process.

**IV. Competitive Analysis**

**4.a. Comparison of Software Solution with Existing Approaches**

Existing approaches such as deploying mathematics as static image files makes editing impossible, kills the semantics associated with the expression, blocks the conversion to and from other mathematical formats (such as graphical displays, speech synthesizers, other mathematics typesetting languages, such as TEX) and brings many other inconveniences when the aim is portability with associated semantics. Deploying mathematics on web pages (or even on regular documents - requires MathML capable word processing software) should be done using MathML. One of the main goals to make the MathML markup language successful is to create mathematical expression editors and converters, which should be designed and developed to facilitate the creation of web pages containing MathML expressions. No matter how successful MathML may be in achieving its goals as a markup language, it is clear that MathML will only be useful if it is implemented well. As a matter of fact this implementation can be best done with software like WME MathEdit. Currently there are several commercial or free MathML software available to achieve part of this goal. The reason why they achieve only “part of” for now is the fact that they are either MathML creators without editing capabilities or they do not work online or at a reasonable speed. WME MathEdit is being developed to be an online MathML creator and editor application which serves students and teachers
who use the WME system to learn and teach mathematics.