

Design, Implementation, and Processing Support of MeML

David Chiu Yi Zhou Xiao Zou Paul S. Wang*
Institute for Computational Mathematics
Kent State University
Kent, Ohio 44242-0001, USA

July 23, 2003

Abstract

The Mathematics Education Markup Language (MeML) is central to the Web-based Mathematics Education (WME) [12] framework (see Figure 1) under investigation at ICM/Kent. Defined in XML syntax, MeML provides effective and expressive markup elements to represent and structure mathematics education pages that may also contain XHTML and MathML [6] elements. MeML aims to empower mathematics teachers, courseware developers, and dynamic mathematics computation and education service providers. And through support of static and dynamic content, MeML promises to deliver an unprecedented mathematics learning environment to students and educators. Unlike most other XML languages, MeML is much more than a domain-specific text structuring and formatting tool. MeML can also access and interact with independently developed and deployed mathematics education content and support services (WME services) anywhere on the Web. Online student-teacher interaction is supported by a WMEchat server that uses MeML to handle mathematical discussions.

1 Introduction

The Mathematics Education Markup Language (MeML) aims to provide an effective and expressive means for authoring and delivering mathematics education content on the Web (Figure 1). MeML is at the core of the WME framework (Figure 2) [13] under investigation at ICM/Kent. MeML works with other components in the WME framework to deliver dynamic and interoperable mathematics education content and diverse online services to students on the Web. The requirements for MeML can be listed as follows.

- Delivering static content, dynamically generated content, and live interactions for learning and experimenting with mathematics on the Web

*Work reported herein has been supported in part by the National Science Foundation under Grant CCR-0201772 and in part by the Ohio Board of Regents Computer Science Enhancement Funds.

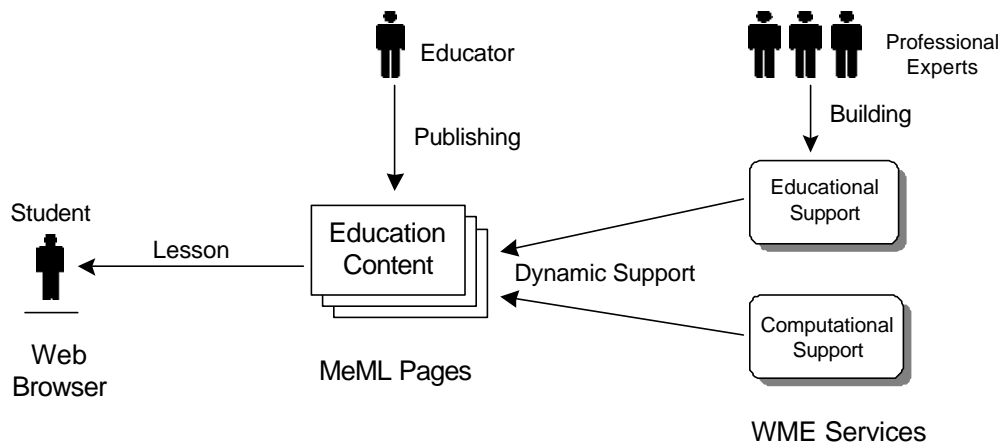


Figure 1: Web-based Mathematics Education

- Providing good support for structuring and organizing mathematics education content
- Being compatible with XHTML and MathML for intermixed usage
- Allowing marked-up pages to be delivered to regular Web browsers for end-user viewing and interaction
- Enabling access and interaction with independently developed and deployed mathematics computation and mathematics education services anywhere on the Web
- Supporting easy cross-referencing and exchange of marked-up content
- Being easy to learn and use, and supplying expressive power for authoring mathematics education materials
- Being flexible and easily extensible for adding features and future growth

Thus, MeML is designed to facilitate authoring, delivering, storing, processing, and exchanging mathematical educational materials over the Web, including live interactions of students with on-Web computational and educational services.

We design MeML as an XML-based markup language that contains enough domain knowledge of mathematics computation and education to be efficient and effective. MeML elements are designed to structure teaching materials as well as to specify learning and educational activities such as exploring the meaning of concepts, putting ideas to work, doing exercises, taking tests, and evaluating results.

Additional MeML elements support accessing on-Web services and live interactions of the student with such services. A variety of educational services can be made available such as example (counter example) generation, answer verification, intermediate steps production, illustration generation, word problem and test generation, test grading, and student performance evaluation.

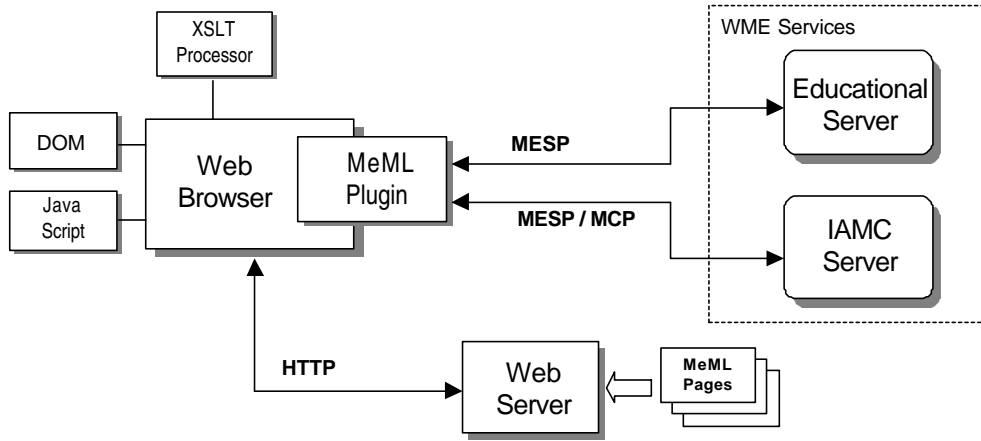


Figure 2: WME Architecture

MeML can deliver static and dynamic mathematics education content as well as live interactions of students with server-side programs to enable experimentation and exploration. Further, we also use MeML in a WMEchat service to enable online student-teacher interaction.

2 Related Work

There were nearly 500 XML-based languages [3] defined by 2000 since the XML recommendation was released by W3C in 1998. We many XML-based standard languages, especially those technologically related to MeML: TEI-Lite [1], DocBook [11], ISO 12083 [7], LMML [5], MathML, OpenMath [4], OMDoc [8], MathBook [2] and SCORM [18].

TEI-Lite is used to define the online books for SGML documents. DocBook and ISO 12083, like TEI-Lite, are also DTDs (Document Type Definition) for structured books. They provide comprehensive ontology and structures for online books. LMML provides DTDs for Web-based computer-science-related learning documents. MathML provides two sets of DTDs for encoding mathematical expressions. One is for the presentation encoding, and the other is for the content encoding. OpenMath is another DTD representing mathematical objects and a set of symbols (content dictionary). OMDoc is an extension of OpenMath. OMDoc provides the DTD for mathematical documents. MathBook is intended to be a DTD simplified from DocBook and OMDoc for Web-based math learning. Although none of these XML-based languages come anywhere close to satisfying the MeML requirements, they do provide good references and DTD models for the MeML work.

3 Overview of MeML Elements

Following XML, we create the MeML DTD [15] that defines the syntax of MeML elements. By examining current textbooks, existing computer-aided mathematics education software, NCTM¹ Principles and Standards, online materials for learning mathematics, and by consulting experts on mathematics education, we can carry out the design of MeML in a spiral approach.

Currently, we have about 46 MeML elements to support the various markup needs for mathematics education. The elements can be divided roughly into five categories: *content elements*, *education elements*, *organization elements*, *computation elements*, and *interaction elements*.

- Content elements—for units of knowledge. Such tags include `<concept>`, `<skill>`, `<terminology>`, `<expression>`, `<variable>`, `<identity>`, `<equation>`, `<diagram>`, and `<theorem>`.
- Education elements—for objectives in education. Tags include `<lesson>`, `<example>`, `<exercise>`, `<homework>`, `<test>`, `<assessment>`, `<diagnosis>`, and `<remediation>`. Contents for these elements can be supplied explicitly or can be generated.
- Organization elements—for page content organization. These include `<abstract>`, `<syllabus>`, `<roster>`, `<guide>`, `<hint>`, and `<summary>`.
- Computation elements—for explicitly invoking WME services to generate dynamic content. Such tags include `<computation>`, `<mathgraph>`, and `<wme>`.
- Interaction elements—for receiving and processing user input. These mainly include the `<interaction>` and `<userinput>` elements.

CSS (Cascading Style Sheet) rules and Javascript code can be associated with MeML pages in exactly the same way as in XHTML.

4 A Complete MeML Page

To get a feel of what an MeML page looks like, let us see an example that shows a complete MeML page. This example teaches double angle identities in trigonometry.

The page is constructed to show MeML constructs rather than how to teach mathematics. The lesson starts with five identities (lines A-E), an explanation for them (lines F-G), and an example with step-by-step solutions (lines H-I).

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="meml.xsl"?>
<!DOCTYPE meml SYSTEM
  "http://cs.kent.edu/~yizhou/MeMLDTD/MeML.dtd">
<meml title="Trigonometry">
```

¹the National Council of Teachers of Mathematics, USA

```

<lesson>
<lesson.content title=
  "Trigonometry: Double-Angle Identities">
<html:ul id="dai" xmlns:html=
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html:li><identity><expression encoding="infix">
  sin(2*\theta) = 2*sin(\theta)*cos(\theta)          (A)
  </expression></identity></html:li>
<html:li><identity><expression>
  cos(2*\theta)=cos(\theta)^2-sin(\theta)^2        (B)
  </expression></identity></html:li>
<html:li><identity><expression>
  cos(2*\theta) = 1 - 2*sin(\theta)^2             (C)
  </expression></identity></html:li>
<html:li><identity><expression>
  cos(2*\theta) = 2 * cos(\theta)^2 -1            (D)
  </expression></identity></html:li>
<html:li><identity><expression>
  tan(2*\theta)=2*tan(\theta)/
  (1-tan(\theta)^2)                               (E)
  </expression></identity></html:li>
</html:ul>
<explanation>                                     (F)
<html:p>An identity is an equality that holds for all
  possible (allowed) values of the variable. In the
  above identifies the variable is
  <variable>\theta</variable> and it can take on
  all real values. ... </html:p>
<html:p>By substituting
  <expression>\theta/2</expression> for
  <expression>\theta</expression> on both sides of
  an identity you can derive<html:em>half-angle
  identities</html:em>... </html:p>
</explanation>                                   (G)
<example title="Using the double-angle identities"> (H)
<problem>
<html:p>Find <expression>sin(2*\pi/3)</expression>
  by using <html:a href="#dai">double-angle
  identities</html:a>.</html:p>
</problem>
<solution>
<step><expression>
  sin(2*\pi/3)=2*sin(\pi/3)*cos(\pi/3)
  </expression></step>
<step><expression>

```

```

        = 2*(sqrteroot(3)/2)*(1/2)
    </expression></step>
<step><expression>
    = squareroot(3)/2</expression></step>
</solution>
</example> (I)
</lesson.content>
<lesson.ender>
<exercise type="interactive" (J)
    id="exerciseTrigonometryIdentities">
    <computation id="exer1"
        wmeurl="http://icm.kent.edu/cgi-bin/trig.wme">
        <parameter name="operation">verify</parameter>
        <parameter name="expr">
            <expression>sin(4*\pi/3)</expression></parameter>
        <parameter name="ans">
            <userinput type="expression"
                name="func" encoding="infix" /></parameter>
        </computation>
    <computation id="exer2"
        wmeurl="http://icm.kent.edu/cgi-bin/trig.wme">
        <parameter name="operation">verify</parameter>
        <parameter name="expr">
            <expression>cos(\pi/4)</expression></parameter>
        <parameter name="ans">
            <userinput type="expression"
                name="func" encoding="infix" /></parameter>
        </computation>
    </exercise> (K)
<homework emailto="teacher@school.edu" (L)
    due="08/08/2003 10:00AM"
    id="homeworkTrigonometryIdentities">
    <do id="assignment">Simplify the expression
        <expression>2*sin(\pi/5 - \pi/2)^2 - 1</expression>
    </do>
</homework> (M)
</lesson.ender>
</lesson>
</meml>

```

At the end of the lesson, you can find two exercises (lines J-K) and a homework with a due date (lines L-M). The exercise can be done online and the student input is sent to a designated WME service, `trig.wme`, for verification. The homework is submitted to the teacher by email. Figure 3 shows the browser display of this MeML page.

Trigonometry - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://www.cs.kent.edu/~yizhou/example/identity.mml> Go

Lesson : **Trigonometry: Double-Angle Identities**

$$\sin(2\theta) = 2\sin(\theta)\cos(\theta)$$

$$\cos(2\theta) = (\cos(\theta))^2 - (\sin(\theta))^2$$

$$\cos(2\theta) = 1 - 2(\sin(\theta))^2$$

$$\cos(2\theta) = 2(\cos(\theta))^2 - 1$$

$$\tan(2\theta) = \frac{2\tan(\theta)}{1 - (\tan(\theta))^2}$$

explanation

An identity is an equality that holds for all possible (allowed) values of the variable. In the above identities the variable is θ .

By substituting $\theta/2$ for θ on both sides of an identity you can derive *half-angle identities*...

example **Using the double-angle identities**

problem Find $\sin(\frac{2\pi}{3})$ by using [double-angle identities](#).

solution

step - 1: $\sin(\frac{2\pi}{3}) = 2\sin(\frac{\pi}{3})\cos(\frac{\pi}{3})$

step - 2: $= 2(\frac{\sqrt{3}}{2})(\frac{1}{2})$

step - 3: $= \frac{\sqrt{3}}{2}$

Lesson Ender :

exercise

1. Verify $\sin(4\frac{\pi}{3})$ Fill in your answer

2. Verify $\cos(\frac{\pi}{4})$ Fill in your answer

homework **Due: 08/08/2003 10:00AM**

1. Simplify the expression $2\sin(\frac{\pi}{5} - \frac{\pi}{2})^2 - 1$

Fill in your answer

Figure 3: Identities in Trigonometry Page

In this example, the very first line is the XML processing instruction specifying the XML version (1.0), and the character encoding (UTF-8). The stylesheet tag indicates the XSLT to use for processing this page. The DOCTYPE element declares the MeML DTD to which this MeML page conforms. The `meml` is the root element. Here, it contains one child element *lesson* which encloses the *lesson.content* and the *lesson.ender* elements.

The *lesson.content* has a title *Trigonometry: Double Angle Identities*. The five entities are listed using regular xhtml elements `ul` and `li`. Allowing mixed use of XHTML and MathML elements in an MeML page makes MeML compact and focused on mathematics education.

Each identity is given by an infix `<expression>`. The notations `\theta` and `\pi` specify Greek symbols. Alternatively, expressions can be given in MathML.

5 MeML Page Processing Overview

Before an MeML page can be displayed by a Web browser, it must be translated into a regular web page. The translation can be performed by a WME Page Processor on the server side or the client side [13]. Currently, we think the client side page processor solution has more advantages. Figure 4 shows the processing steps for an MeML page.

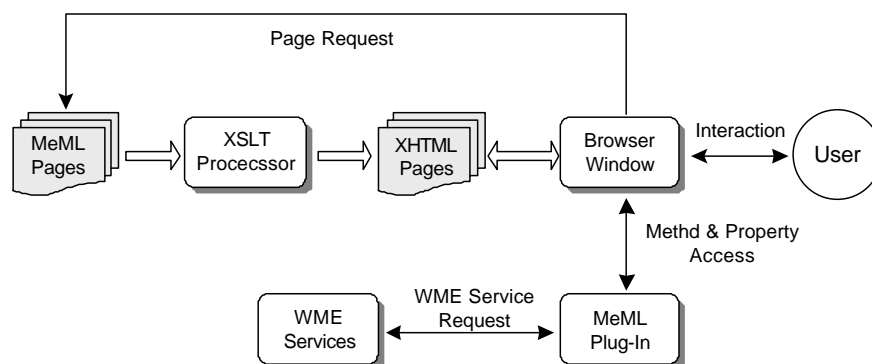


Figure 4: Accessing MeML pages

A typical WME Page Processor is composed of:

- An XSLT processor and XSLT templates
- A collection of Web browser plug-ins called MeML plug-ins

An MeML page may contain XHTML and MathML in addition to MeML tags. Regular Web browsers do not have the ability to process MeML elements directly. The MeML elements must first be translated into XHTML and MathML. Mathematical expressions are translated into MathML and displayed either by MathPlayer [17] or directly by the MathML-enabled Web browser. The translation can be done by using an XSLT processor. The translated XHTML page may also have embedded *Event Handlers* (in JavaScript) and references to MeML plug-ins. The translated page can be processed by a regular Web browser.

When a Web browser loads the translated page, references to MeML plug-ins will be instantiated as in-page objects. Depending on the type of browser, these object instances may take the form of ActiveX Controls (in Internet Explorer) or Plug-ins (in Netscape Navigator). Some important MeML plug-ins are *MathComputation*, *MathGraph*, and *MathExpression*. When a user interacts with the page, a series of in-page events may be generated from input-control elements (such as XHTML `<input>` elements) and MeML plug-ins (visible or hidden). The event handlers treat these events by interfacing to appropriate MeML plug-ins to support the functionality of specific MeML tags.

Woodpecker, a prototype client-side WME Page Processor, is under development at Kent/ICM. Figure 5 shows the architecture of Woodpecker. We will use examples to show how Woodpecker handles MeML tags.

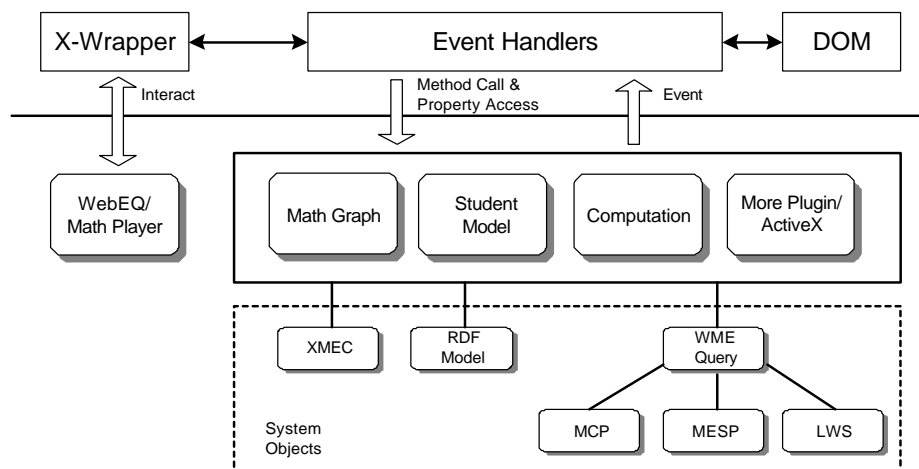


Figure 5: Woodpecker Design

6 MeML Element Processing Example

Let's illustrate the MeML page processing procedure by examples. We'll begin by looking at the `<computation>` element in detail.

The `<computation>` tag provides a way to invoke any WME service, identified by an URL (line 1), for any particular *operation* (line 2) supported by that service. Arguments required by the particular operation, expand in this case, are given with the `<parameter>` element (line 3).

```
<computation id="Compu1"
  wmeurl="mcp://symbolicnet.org/polynomial.wme" > (1)
  <parameter name="operation">expand</parameter> (2)
  <parameter name="polynomial"> (3)
    <expression id="expl" encoding="infix">
      (x+1)*(x-1) </expression>
  </parameter>
```

```
</computation>
```

If the expression `exp1` has been defined earlier in the MeML page, it does not need to be repeated and the notation

```
<use name="exp1" />
```

can be given instead.

Here is part of the XSLT code for `<computation>` tag:

```
<xsl:template match="computation">
  <object>
    <xsl:attribute name="id">
      <xsl:value-of select="@id" />
    </xsl:attribute>
    <xsl:attribute name="classid"><xsl:text>
      clsid:4C7E7765-8C39-11D7-A830-0009457281D3
    </xsl:text>
    </xsl:attribute>
    <param name="wmeurl">
      <xsl:attribute name="value">
        <xsl:value-of select="@wmeurl" />
      </xsl:attribute>
    </param>
    <param name="operation">
      <xsl:attribute name="value"><xsl:value-of
        select="parameter[@name='operation']" />
      </xsl:attribute>
    </param>
    <xsl:for-each select=
      "parameter[not(@name='operation')]">
      <param><xsl:attribute name="name">
        <xsl:value-of select="@name" />
      </xsl:attribute>
      <xsl:attribute name="value">
        <xsl:text>ID{</xsl:text>
        <xsl:if test="self::node()[use]">
          <xsl:value-of select="node()/@name" />
        </xsl:if>
        <xsl:if test="self::node()[expression]">
          <xsl:value-of select="node()/@id" />
        </xsl:if>
        <xsl:text>}</xsl:text>
      </xsl:attribute>
    </param>
  </xsl:for-each>
```

```

    </object>
    <xsl:apply-templates
      select="parameter/expression" />
</xsl:template>

```

After the first processing step of WME Page Processor, the `<computation>` tag will be translated into the following result.

```

<object id="Computel" classid=
  "clsid:4C7E7765-8C39-11D7-A830-0009457281D3">
  <param name="wmeurl" value=
    "mcp://symbolicnet.org/polynomial.wme" />
  <param name="operation" value="expand" />
  <param name="polynomial" value="ID{exp1}" />   (4)
</object>

```

The polynomial $(x+1) * (x-1)$ is transmitted to the *MathComputation* plug-in (line 4) via the element id through DOM.

Thus, the `<computation>` tag is mapped to an instance of MeML *MathComputation* plug-in which is a invisible object. The `classid` points to the plug-in's registration ID. A group of JavaScript event handlers, which are bound with the plug-in, are also inserted into the translated page. We will not show all event handlers in this paper. But here is a section missing from the above XSLT code to help comprehension.

```

<xsl:template match="computation">
  <script type="javascript"
    event="AccessDOMObject(ObjectID, ObjectType, ParamName)">
    <xsl:attribute name="for">
      <xsl:value-of select="@id" /></xsl:attribute>
    <xsl:comment>
      var OID = getElementById(ObjectID);
      <xsl:value-of select="@id" />.
      SetParameter(ParamName, OID.value);
      /* Other JavaScript Code ... */
    </xsl:comment>
  </script>
  <!-- Other XSLT code to generate <object> -->
</xsl:template>

```

This XSLT transformation will generate following event handler for our `<computation>` example:

```

<script type="javascript" for="computel" event=
  "AccessDOMObject(ObjectID, ObjectType, ParamName)">
  <!--
    var OID = getElementById(ObjectID);

```

```

        compute1.SetParameter(ParamName, OID.value);
        /* Other JavaScript Code ... */
    -->
</script>

```

After the translated page is loaded into Internet browser, this MeML plug-in instance will fire an event `AccessDOMObject()` to the browser to obtain `exp1`. The event handlers will catch the event, retrieve the expression's text value through Document Object Model, and send the value back to the MeML plug-in. With the expression, MeML plug-in will send a MESP request to the remote compute service, a *WME Service* [13] using the specified URL (line 1). When the computing result is returned from the WME service, the MeML plug-in will fire another event `UpdateDOMObject()`. Finally, event handlers will update the display of expression with the new value through DOM again. The `<computation>` tag can also specify target elements in the page where the dynamically computed results will be inserted into the page. With this kind of round-trip between Internet browser and remote WME Services, MeML page can generate rich dynamic content under the support of WME Page Processor.

The mechanism for the `<mathgraph>` and `<expression>` tags is similar to that for the `<computation>` tag. But these tags have very different functions. The `<mathgraph>` tag is mapped to the MeML plug-in *MathGraph*, which is a visual object in the Web page, and *MathGraph* generates 2D, 3D figures in its allocated area in the browser window.

The `<expression>` tag is mapped to another hidden MeML plug-in *MathExpression*. It is responsible for manipulating all in-page mathematical expressions. It also contains an implementation of XMEC (eXtensible Mathematics Encoding Converter) [14]. It will convert in-page expressions among four primary encodings: MathML, MP, Infix, and OpenMath. This capability is also handy when calling WME services or generate appropriate display in the browser.

The events that *MathGraph* and *MathExpression* can fire are similar to the *MathComputation* plug-in. But they have different event handling methods for interacting with the Web browser.

7 Student-Teacher Interaction

So far, we described providing mathematics education with MeML pages and server-side support for these pages. Now, we turn our attention to another important aspect: student-teacher interactions while viewing MeML pages. It is important that learners be able to interact with each other, with resources of instruction, and with their teacher [10].

7.1 The WMEchat Model

We introduce a Web-based chatroom model, simply called WMEchat, to support this need. Generally, WMEchat provides a classroom model for student-teacher interaction using the Web as the delivery medium for the chat.

In the simplest case, a WMEchat server receives input from participants in the same chatroom and keeps a running log of that input in a file that takes the form of an MeML page. This file is viewed by each chatroom participant with a common Web browser enabled by the Woodpecker plug-in.

We choose to implement WMEchat as a Web service for the following reasons.

- Ease of displaying mathematical expressions—With MathML support in Internet browsers, we easily overcome the difficulty of mathematical display.
- Simple integration—The WMEchat can be deployed along-side MeML pages by a server host.
- Easy database access—Many CGI scripts and server-side languages natively support database access. A database can provide much support for the WMEchat server and for registered users.
- Support of Mathematics—By using MeML, we can better support chat on mathematics.
- Extensibility—Chat related WME services can augment and extend the basic functionalities and features of WMEchat.
- Security—Classroom chat is easily encrypted through SSL.

7.2 Implementing WMEchat

Before we start discussing implementation, please note that some notions are based on the IRC protocol [9], namely the administrative functions, and the status of users; both of which will be discussed later. Naturally, because of this association, we also chose to inherit some of the IRC terminology. Below is a list of conventions that we will use throughout this section.

- Chat rooms will be referred to as classrooms.
- Classroom administrators will be referred to as operators.
- The user establishing a classroom is known as the founder.
- To *op a user* means to grant the user operator status.
- To *voice a user* means to give the user *voiced* status which allows the user to talk instead of just listen.
- To *kick* means to exclude a user from the classroom.
- To *ban* means to exclude a user from the classroom temporarily until the ban is lifted.
- Classroom operators are denoted by a @ symbol before their nickname in the user list.
- Voiced users are denoted by a + sign before their nickname in the user list.

Figure 6 and Figure 7 show the WMEchat architecture and a magnified view of a single chat room respectively.

A chat window is an instance of a browser window. It consists of three separate areas: dialogue frame, user list frame, and input frame.

A user talks and enters WMEchat commands through the input frame. The chat messages can be a mixture of plain text, XHTML, MathML, and MeML. This message is sent to the Web server via HTTP. The chat server processes incoming messages sequentially. Talk messages are added as

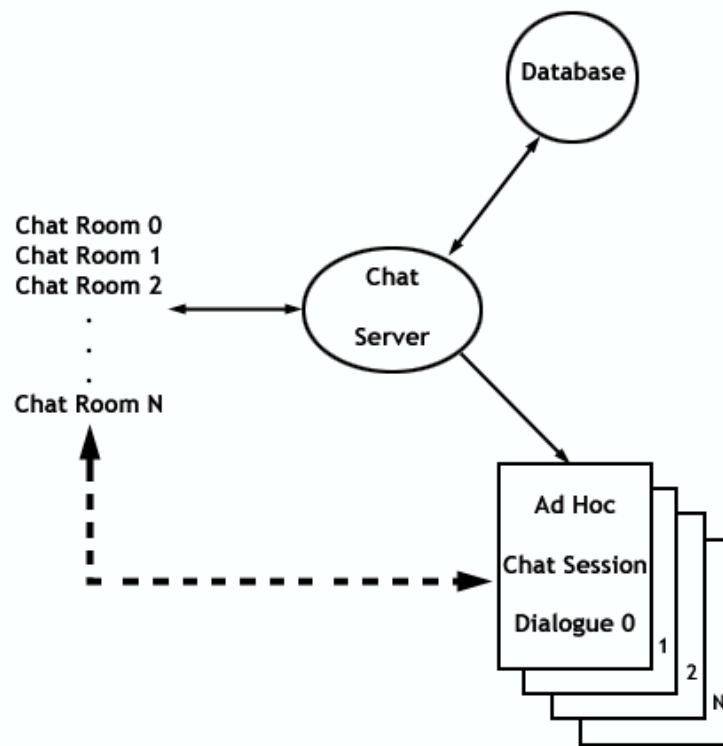


Figure 6: WMEchat Architecture

markup elements to a dialogue file for the particular classroom. This dialogue file is an MeML page displayed in the dialogue area. This process repeats until the classroom is closed, at which point the dialogue file becomes a classroom chat log, identified by date/time, available for downloading any time later. A classroom can have multiple chat sessions, in other words, a teacher can reuse the class on a different day or time.

The current implementation refreshes the dialogue frame every five seconds and the rate is a WMEchat server configuration parameter. The user list frame displays the nicknames of everyone that has joined the classroom and provides a GUI for operator controls (op, voice, kick, and so on) of the users. The input frame contains an input box and a send button. The input box is made large enough for users to type paragraphs and lengthy MathML, XHTML, or MeML element blocks. Because MeML supports mixing of XHTML and MathML elements we can say that the user may enter a valid MeML block.

Let's now consider the administrative and security aspects for WMEchat.

7.3 WMEchat Portal and Security

A WMEchat portal refers to a *control panel* Web page. There are two types of portals, one for each type of user: the teacher and the student. The teacher's portal consists of the following tools and options:

User Functions

- Login and Logout

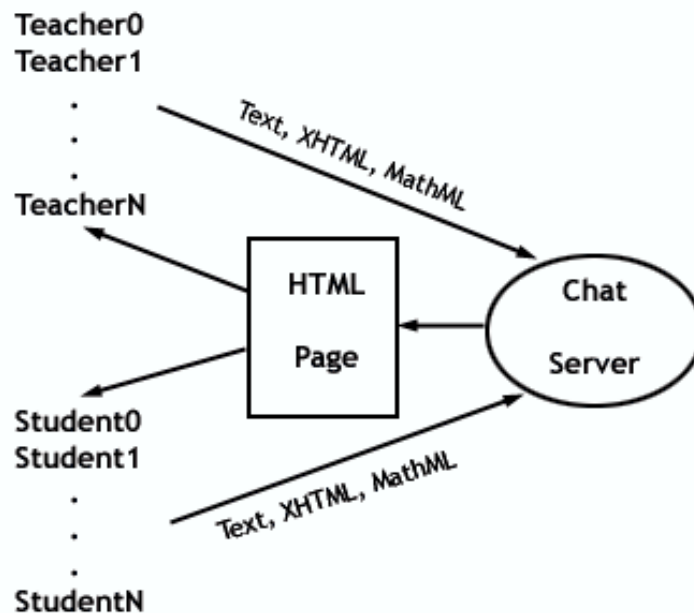


Figure 7: A Single WMEchat Room

- Change password
- Change email address
- Retrieve password

Classroom Functions

- Create a new classroom
- Reactivate a classroom
- Retrieve classroom listing
- Retrieve classroom log

A teacher is able to access all portal controls, while a student is restricted to user functions and classroom listing.

The first form of security is registration. By default, no public entrance into any classroom is allowed. This is to prevent easy unauthorized entrance from the general public and potentially malicious users. Every user must be *registered* with the WMEchat in order to partake in a classroom discussion.

Student registration is performed by the classroom founder, the teacher in most cases. The founder creates new users by supplying WMEchat with their email address, nickname, password and other optional information such as first and last names. Registered users can be associated with one or more classrooms. After login, students may change their user information and password as stated before. Initially a student's only privilege is to join the assigned classroom. A WMEchat service initially contains a root teacher who is able to create other teachers.

The classroom founder may also control settings for the classroom. The settings include limits on total number of participants, anonymous joining, and dialogue moderation. Once set, these attributes stay with a classroom until they are changed.

- A *limit* is simply the number of users that are allowed to join, including the founder. If the limit is not set, then there is no limit.
- *Anonymous_entrance* allows un-registered users to join the classroom. These users will not have the ability to speak. The impetus for anonymous entrances is the acknowledgement that users may want to get an idea of the classroom before registering for it.
- A classroom is *moderated*, only operators and voiced users can talk. Other users may request to be voiced and an operator may voice/silence users.

Operators has many privileges including: speak during moderation, toggle moderation set user limit, kick, ban, op, and voice another user, as well as end the chat room session.



For convenience, the classroom invocation page supplies two modes: classroom and general discussion.

If the classroom mode is selected, moderation is automatically chosen. This concept models real-life classrooms. Normally, the teacher lectures, and the students raise their hand to request to speak. In the classroom mode, normal users must make a voice request, similar to raising a hand, to obtain the right to speak. A voice request is denoted by a graphical hand, next to a user's nickname, followed by a sequence number indicating who is first, second, in making the voice request. An operator may click on the hand graphic to voice/silence the user. In the general discussion mode, moderation is toggled off. This allows everyone in the classroom to speak as a default.

Once a chat session ends, the classroom is marked inactive which prevents other users from joining until reactivation by the teacher. This is to prevent a "takeover" by other users that may not be initially affiliated with the classroom. Classroom reactivation does not require the teacher to perform student registration again. This information is stored in the WMEchat database which is used to support the WMEchat service. The details of the database organization are out of the scope of this paper.

8 WMEchat Example

Figure 8 shows WMEchat in action. Here are some clarifications and things to look for:

- Notice that the classroom settings, such as moderation, is shown on the title bar.
- Notice that the operators are denoted with  and voiced users with  as mentioned previously.
- Line 3 in the dialogue window is an example of MathML display.

The following is the MeML code for the dialogue window. Notice that MathML and XHTML elements can be interlaced with MeML.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE meml SYSTEM "C:\MeML\MeML DTD\MeML.dtd">
```

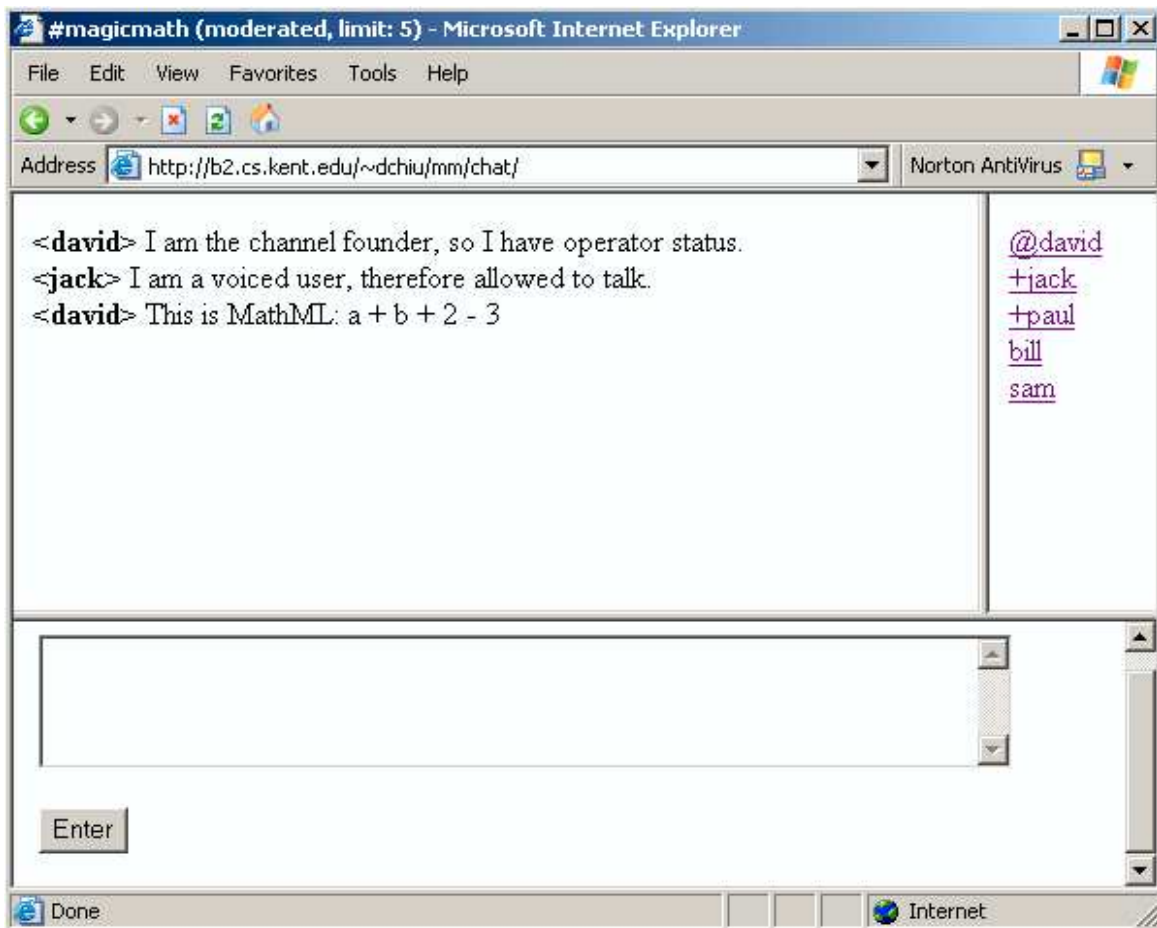



Figure 8: A Typical Chat Window

```

<?xml-stylesheet type="text/xsl" href="mem1.xsl"?>
<mem1>
<html:head>
  <meta http-equiv="Refresh"
        content="5;url=chat.php" />
  <meta http-equiv="Content-Type"
        content="text/html; charset=iso-8859-1" />
</html:head>
<html:body>
<b>&lt;math>I am the classroom founder, so I have
operator status.</math></b><br />
<b>&lt;math>I am a voiced user, therefore allowed
to talk.</math></b><br />
<b>&lt;math>This is MathML:
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mrow>
    <msup><mrow>

```

```

    <mi>a</mi><mo>+</mo><mi>b</mi>
</mrow></msup>
<mo>+</mo>
<msub>
  <mrow><mi>2</mi> <mo>-</mo><mi>3</mi></mrow>
</msub>
</mrow>
</math>
</html:body>
</mhtml>

```

MathML is interlaced directly with XHTML in order to achieve the mathematics display, instead of using images or infix . This lets complex mathematical equations to be displayed in the browser and allows the educator to teach mathematics without the hindrance of display.

9 Future Work

We have reported on our progress and on-going efforts in MeML and in student-teacher interaction support with the WMEchat utilizing MeML.

There is much work ahead. The set of MeML elements and their functions will continue to evolve to better support quality teaching and learning of mathematics. The software components of the WME framework, of which the MeML language is a part, will continue to be tested and refined. Better GUI's can make MeML and WMEchat easier to use and more effective. A client-side plug-in for the WMEchat service can eliminate the wasteful constant page refreshing and support a display as necessary model for the chat session. Elements designed to invoke WMEchat can be very useful inside an MeML page.

References

- [1] Lou Burnard. . *An Introduction to Text Encoding for Interchange* www.tei-c.org/Lite
- [2] H. Cuypers and H. Sterk. *Mathbook, web-technology for mathematical documents*. Proceedings of the BITE 2001 conference. See also from www.riaca.win.tue.nl.
- [3] Berthold Daum, and Udo Merten. *System Architecture with XML*, Morgan Kaufmann Publishers.
- [4] Mike Dewar (Web Master). *OpenMath*. The OpenMath Society, www.openmath.org/.
- [5] Burkhard Freitag. *Learning Material Markup Language*, www.lmml.de.
- [6] Max Froumentin, Team Contact for the Math Working Group. *MathML*. www.w3.org/Math
- [7] Dianne Kennedy. *ISO 12083*. www.xmlxperts.com/12083.htm

- [8] Michael Kohlhase. *OMDoc: Towards an Internet Standard for the Administration, Distribution and Teaching of mathematical Knowledge*. Proceedings of Artificial Intelligence and Symbolic Computation Springer LNAI 2000.
- [9] J. Oikarinen and D. Reed “Internet Relay Chat Protocol” (RFC 1459).
- [10] Simonson, M., Smaldino, S, Albright, M., and Zvacek, S. (2003). *Teaching and learning at a distance: Foundations of distance education (2nd ed.)*. Upper Saddle River, NJ: Merrill.
- [11] Norman Walsh. *DocBook XML 4.2*, www.oasis-open.org/docbook/xml/4.2/index.shtml
- [12] Paul S. Wang, Norbert Kajler, Yi Zhou, and Xiao Zou. *Initial Design of A Web-Based Mathematics Education Framework, Proceedings Internet Accessible Mathematical Computation 2002 Workshop*, Lille, France.
- [13] Paul S. Wang, Norbert Kajler, Yi Zhou, and Xiao Zou. *WME: Towards a Web for Mathematics Education*, Proceedings of ISSAC 2003 (to appear).
- [14] Xiao Zou and Paul Wang, “XMEC: An Extensible Mathematical Encoding Converter”, <http://icm.mcs.kent.edu/research/xmec/>
- [15] Yi Zhou. *MeML DTD*. <http://www.mcs.kent.edu/~yizhou/MeMLDTD.pdf>
- [16] Yi Zhou. *MeML Specification*. <http://www.mcs.kent.edu/~yizhou/Draft of MeML Specification.pdf>
- [17] MathPlayer and WebEQ, www.mathtype.com.
- [18] The Advanced Distributed Learning Initiative, Office of the Secretary of Defense. *Sharable Content Object Reference Model*. www.adlnet.org/
- [19] McDougal Littell, Inc. *Sample of Program Components*.