

The Design of A Web-Based Mathematical Education Framework

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

February 7, 2002

Abstract

An approach for Web-based Mathematics Education (WME) is presented. WME aims to establish a framework to support authoring of dynamic and interactive mathematics education pages, end-user access through regular Web browsers, and interoperable back-end *WME servers*. The WME architecture builds on emerging Web standards, and defines an XML-based authoring language, the *Mathematical Education Markup Language* (MeML) supported by a server-side MeML interpreter. WME servers are interoperable by conforming to a common interface specification.

1 Introduction

Web-based learning can extend the reach of education and significantly broaden its impact and influence. There are many approaches but few are effective for teaching/learning mathematics. Given the state of mathematics education in the United States and other countries, an effective way to deliver teaching and learning materials over the Internet/Web holds much promise.

While various methods have been used to display mathematical formulas in Web pages and to make simple mathematical computations accessible via CGI programs or X Windows [16], a general and effective system for accessing, producing and delivering mathematical content is still the subject of research and development.

Investigators at the W3 Consortium (W3C) and elsewhere are working to make *publishing* mathematical materials on the Web easy. *MathML* [20] defines an XML language for markup of mathematical expressions with support for both presentation encoding (display layout) and content encoding (computation semantics).

*Work reported herein has been supported in part by the National Science Foundation under Grant CCR-9721343 and in part by the Ohio Board of Regents Computer Science Enhancement Funds.

The IBM digital publishing group has released the experimental *Techexplorer* [15], a Web browser plug-in that dynamically formats and displays documents containing scientific and mathematical expressions coded in $\text{T}_{\text{E}}\text{X}/\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$. Some MathML is also supported. Techexplorer also allows a user to send expressions to a fixed compute server for evaluation. *MathType* [26], from Design Science Inc., supports interactive creation of mathematical notations for web pages and documents. The same company also offers *WebEQ* [26] that provides a Java applet to display WebTeX and MathML in a browser. The W3C *Amaya* Web browser demonstrates a prototype implementation of MathML which allows users to browse and edit Web pages containing mathematical expressions [25]. Together with the rest of the Web page, these expressions are manipulated through a WYSIWYG interface. The increasing acceptance and software support for MathML were evident at the recent *MathML International Conference 2000* [10].

Mathematical content viewing on a Web page is static. On the Internet, end-users, especially educational applications, can make good use of *dynamic access to mathematical computing*. “Internet Accessible Mathematical Computation” has been the subject of the *1999 IAMC Workshop* (part of ISSAC’99) and the *2001 IAMC Workshop* (part of ISSAC’01). The full-day workshops underscore the on-going interest in making mathematical information and computation easily available in the new communication age [2, 5]. For more background and related activities, please refer to the Proceedings of the IAMC Workshops [8], the IAMC homepage [9], and the Workshop on *The Future of Mathematical Communication* [11].

At the Institute for Computational Mathematics (ICM/Kent), efforts have been made to build a *distributed IAMC framework* [6, 4] which can support both interactive and transparent access to mathematical computation on the Internet/Web through the *Mathematical Computation Protocol*.

Researchers have begun to make attempts to deliver mathematical education materials over the Web/Internet. Already, we can find many Web sites providing courses and tools for mathematics education. Such sites include WIMS [28], Livemath [18], Mathwright [24], WebMathematica [27], Calc101 [13], AcitveMath [12], Maple [19], and MathWeb [23]. Among these, WIMS is a well-known site for Web-based mathematical education created by Xiao Gang in France. WIMS uses a case-oriented approach by providing a new CGI program for each new type of educational content page. Generally, ad hoc server-side programming are made to support a narrowly defined set of topics. Authoring educational content within the scope is awkward and outside of the scope is almost impossible without substantial new back-end Web programming. The most serious flaw of such ad hoc approaches is that the components, content pages, and server-side programs do not combine to form a larger system within which to interoperate and to mutually reinforce. Linda Beccerra [1] gave a good summary of Web tools for interactive computation.

It is perhaps time to consider a systematic way of creating and supporting mathematics education on the Web.

We investigate a *Web-based Mathematics Education* (WME) framework based on our experience and results with IAMC [3]. WME works with regular browsers, makes authoring simple and easy, allows systematic access to server-side support, and enables these independently developed components to interoperate seamlessly. In short we sought to create a *Web for Mathematics Education*.

We begin by overviewing the goals of WME. Then, the design and architecture of WME are described. The Mathematical Education Markup Language and its support are considered in some detail.

2 WME Goals

The goal of our research is to produce a framework for mathematics education that allows easy development of

1. mathematics educational content
2. mathematical and other support programming

both accessible and interoperable on the Web/Internet. Thus, content pages and support programming can be developed independently but can interoperate from anywhere on the Web. Ultimately, WME must support mathematical education needs effectively.

The WME framework has these characteristics:

- WME is decentralized allowing everyone to put WME pages on the Web and to supply back-end compute powers that support any WME pages.
- WME combines existing technologies and emerging Web standards in a sensible and innovative way.
- A WME page is as easy to create as an HTML page and can be handled through common Web browsers.
- A WME page can combine static content, dynamically generated mathematical results, and interactive computations for the end user.
- Mathematical computations and interactive educational functions initiated from a WME page can be supported by any server-side resources identified by URLs.

Clearly, for WME to work, it must have content-markup support, front-end support, and back-end support. More importantly, such support must be delivered within an architectural framework that uses appropriate technologies to integrate components, allowing them to interoperate in a seamless manner on the Web.

3 The WME Architecture

Figure 1 shows the WME framework architecture.

WME supports Web-based mathematical education through a combination of technologies.

- *Content-markup support* — Mathematics education contents are represented by Web documents called *WME pages*. A WME page is written in the *Mathematical Education Markup Language* (MeML) (Section 4). Basically MeML allows XML-defined tags (*MeML tags*) to be used together with HTML and MathML tags for easy content markup. MeML includes tags designed to invoke *WME servers*, services conforming to the *Open WME Interface* (OWI).

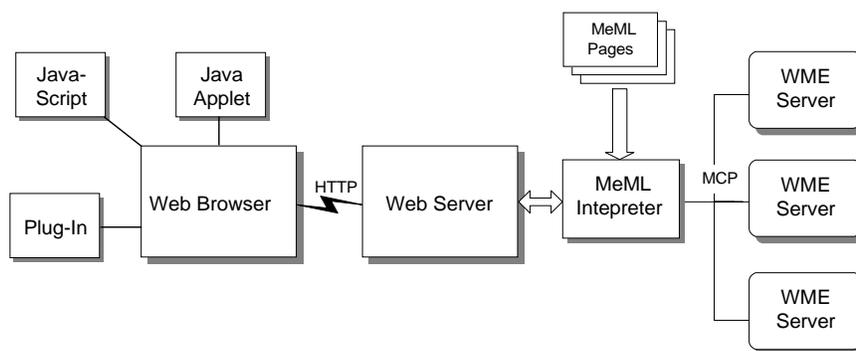


Figure 1: WME Architecture

- *Back-end support* — On the server-side, WME documents are processed by an *MeML interpreter* (as a Web server module) that transforms each MeML page, on access, into a regular Web document suitable for standard browsers. The interpreter invokes WME servers as specified by MeML tags to obtain *embedable content* to be included in the final Web page delivered through the Web server. Content from a WME server may contain MeML which will be processed accordingly. WME servers conform to OWI and are accessible by URL. Such embedable content may supply mathematical expressions, computations, graphical displays, educational functions, and other services. The resulting page has appropriate client-side programming (CSS, Javascript, DHTML, Java Applet, etc.) to support dynamic and interactive learning. Thus, the MeML interpreter and open WME servers constitute the back-end support.
- *Front-end support* — On the client side, common Web browsers with MathML display and editing capabilities can be used. Additional front-end features can be provided through Javascript programming, the *document object model (DOM)*, *Cascading Style sheets (CSS)*, Java applets, and browser plug-ins.

Thus, the WME architecture involves these major components: the Mathematical Education Markup Language (MeML), WME pages written in MeML, the MeML interpreter, the Open WME Interface (OWI), WME servers conforming to OWI, and standard Web browsers.

Usage Paradigm

In the WME framework, an online courseware consists of a group of WME pages with dynamic back-end support by WME servers supplying a rich set of computational and educational functionalities.

With WME, a new paradigm of Web-supported mathematical education emerges:

- Students can access WME pages for a dynamic and interactive learning experience from anywhere on the Web, at anytime. They can study at their own levels following their own paces. Instructional materials may largely come from courses in schools, but the materials can be supplemented by pages from other authors globally.

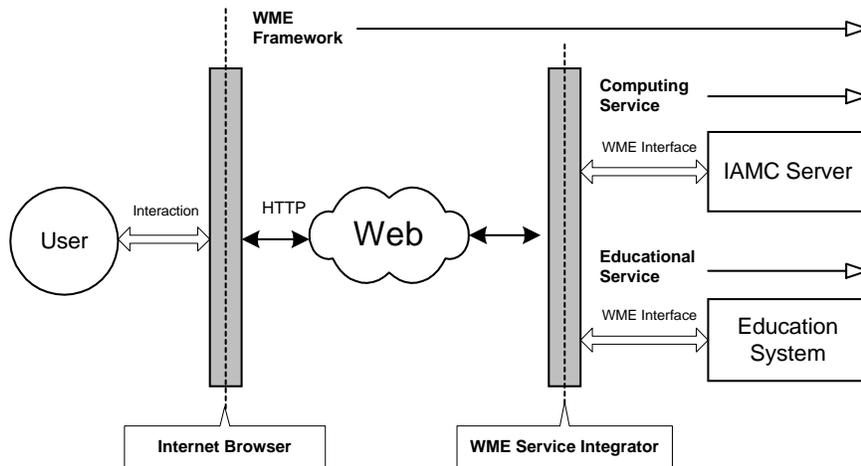


Figure 2: WME Conceptual Model

- Software developers can focus on building useful WME servers. These servers can be used by anyone from anywhere on the Web. Each server can provide well-defined set of capabilities within a particular scope. For example, there may be a *fractions server*, a *polynomial server*, an *algebraic equation server*, a *derivative server*, an *integration server*, etc. As long as the servers conform to OWI, they can be used in any education content pages (in MeML). We also envision a variety of back-end educational services including example (counter example) generation, answer verification, intermediate steps production, plotting/graphing, illustration generation, word problem generation, test generation, and test grading.
- Educators author interactive online courseware as a set of WME pages. A WME page is written in MeML either directly or via authoring tools. Authors can focus on teaching mathematics rather than dealing with the underlying Internet/Web technologies. An author can easily include, mix and match, powerful interactive mathematical education features by accessing WME servers developed by others.
- Educators deploy and maintain their courseware simply and easily. Authors can modify, revise, and change their WME pages anytime from anywhere.
- Server developers can upgrade and improve WME servers without affecting WME pages as long as equivalent contents are returned.

The *MeML Interpreter* translates WME pages into regular Web pages on-the-fly, accessing WME servers specified by MeML elements for dynamically computed content.

Figure 2 shows the WME Framework concept and the overall architecture.

4 Mathematical Education Mark-up: MeML

To support WME page authoring, we are developing the *Mathematical Education Markup Language* (MeML) defined by XML. MeML elements can contain MathML and can be embedded in

HTML pages. MeML will be expressive enough to make authoring Web pages containing mathematics, interactive computations, and educational functions easy and straight-forward. For ease

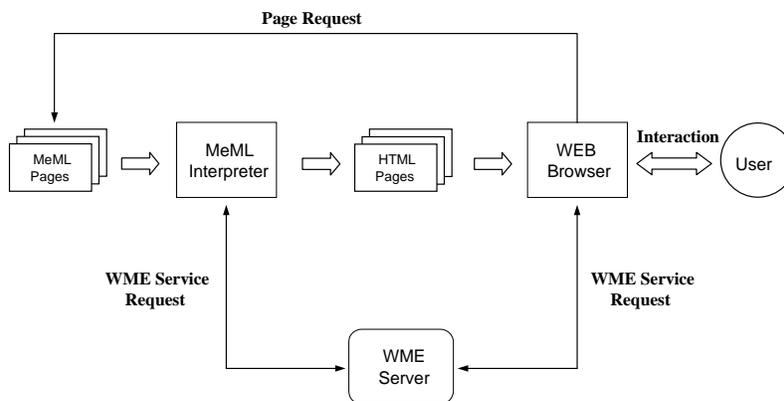


Figure 3: Accessing MeML pages

of use, MeML regards any un-recognized tag to be in HTML and requires MathML tags to be enclosed in the MeML `math` element.

With MeML, an author can write HTML-like code to create mathematics education pages. An MeML page can have both fixed and dynamically generated content. The MeML interpreter controls code generation and will also access WME servers designated in the MeML page to obtain generated results. The MeML page can specify interactions with the end-user (the student) for an interactive learning experience.

WME supports dynamic interactions with the end-user where page updates can be rendered in-place, in a pop-up window, or in a new page. MeML tags support invocation of any WME servers by URLs and allow mathematical expressions to be encoded in MathML content (for computing), MathML presentation (for rendering), or infix (for convenience).

MeML allows infix notation as a convenience for authors and end-users to enter mathematical expressions. Infix expressions are translated into equivalent MathML content encoding automatically within the WME framework. Functions and mathematical elements defined in MathML can be used in the infix notation. Mathematical constants such as π are entered as `PI` and translated into `&pi ;`. Several other simple rules apply.

Basically, there are four types of MeML elements (tags).

- Structural elements – to provide document structural and organization for a page.
- Mathematical computation elements – to generate mathematical results, create mathematical formulas, and supply interactive mathematical computations for the end-user.
- Educational function elements – to automatically generate content for well-defined mathematics education functions such as generating exercises, supplying examples, or giving tests.
- Declaration elements – to provide information to the MeML interpreter and to control how it works.

Structural Elements

MeML structural elements provide the page with structure by identifying the purpose of different parts of the page. They provide a DOM (document object model) organization for WME pages, thus enabling their flexible usage and automated processing.

There can be many structural tags including `lesson`, `lesson-plan`, `prerequisite`, `skill`, `example`, `test`, `exercise`, `example`, `definition`, `expression`, `theorem`, `formula`, `variable`, `table`, `abstract`, `review`, `summary`. The MeML interpreter transforms structural tags into HTML tags for Web browsers to display. The translation rules are part of the XML definition for each MeML element.

For example, a `lesson` element for fraction addition can be written as follows:

```
<lesson title="fraction addition" id="fraction-add"
        goal="learn addition of fractions">
  <skill topic="Fraction" id="fraction-add-skill">
    <step id="fa1"> ... </step>
    <step id="fa2"> ... </step>
    ...
  </skill>
  <example id="fe1"> ... </example>
  <example id="fe2"> ... </example>
</lesson>
```

The `id`'s also enable referencing by hyperlinks. A `lesson` within one MeML page can be included in another page with the code

```
<lesson id="..." url="...">
</lesson>
```

The `literal` element allows authors to include arbitrary character strings to be displayed literally.

Mathematical Computation Element

Mathematical computation elements introduce dynamically computed mathematical contents in WME pages. The elements themselves are not visible by the end-user. Results returned by WME servers are in MeML ready to be converted to front-end objects.

Here are some examples:

- **<pattern>** – Defining a mathematical pattern. A pattern is an input expression with several variables that can be replaced by a range of values. Patterns are used to generate expressions dynamically.
- **<assign>** – Assigning a mathematical object to a variable.

- **<substitution>** – Replacing a part of an expression with another.
- **<verification>** – Testing a logical expression and return a truth value. HTML code to display for the true case and the false case can be specified in the element.
- **<idref/>** – Cross referencing existing expressions, variables, patterns, and computed results.
- **<computation>** – Requesting results from a WME server. The result obtained becomes a valid MeML object identified by the id of the requesting `<computation>` element.
- **<interaction>** – Creating an object for end-user interactions. The object is supported by a specified WME server. The form and content of `interaction` are determined by the target WME server with input data given in the `interaction` element
- **<mathgraph>** – Creating a graphic object from given expressions and variables.
- **<guiaction>** – Defining an active block for the client-side GUI to perform in-page event-driven interactions with the end-user. This feature supplies the interactiveness needed for a dynamic learning experience.

Educational Function Elements

In MeML, *educational function elements* generate objects to perform well-defined educational activities supported by WME servers. For instance, `gen-example`, `gen-exercise` and `gen-test` can generate examples, exercises, and tests simply and conveniently. Furthermore, the materials generated are different each time. Other educational function elements include `assess`, `diagnose`, and `evaluate`.

Let's look at the `assess` element to get a better idea about educational function elements. The element

```
<assess name="addition of fraction" URL=".../basic.wme">
  <vocabulary name="Rational Number"/>
  <vocabulary name="LCM"/>
  <vocabulary name="GCD"/>
  <vocabulary name="numerator"/>
  <vocabulary name="denominator"/>
  <skill name="addition of fractions"/>
</assess>
```

asks the url-specified WME server to generate materials for assessing the given vocabulary terms and skill. The target WME has the required domain knowledge and computational capabilities to generate page contents to interactively assess the comprehension and skills of student users.

To support basic educational functions, a prototype basic WME server will be developed as part of this project to demonstrate feasibility and to perform experiments.

Next, consider the generation of examples.

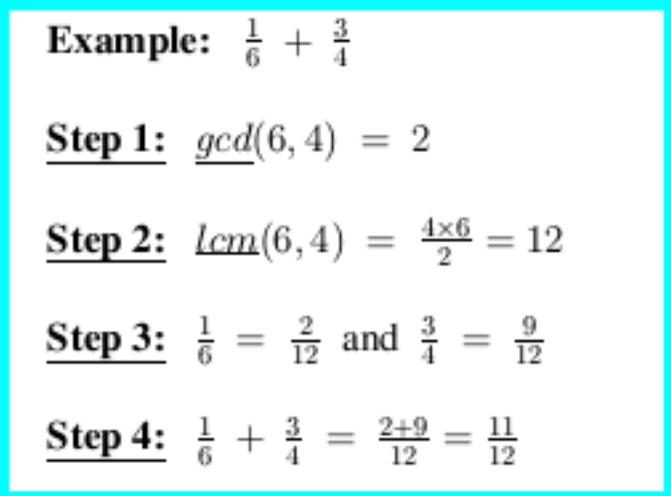
```
<gen-example id="fraction-add-1" topic="fraction" level="basic"
  style="multichoice" count="3" url="../../../basic.wme">
  <idref target="fracpattern"/>
  <interval> <cn type=integer>1</cn>
    <cn type=integer>100</cn>
    <boolean> ... </boolean>
  </interval>
</gen-example>
```

The `gen-example` element specifies a pattern from which to generate the example. A pattern is simply any input expression with *parameters* to be replaced by generated values of numerical types (integer or real) within the given range and satisfying the boolean condition, if any. If there are more pattern parameters than intervals given, the last interval applies to all the remaining parameters.

For example, the fraction-add pattern can be,

```
<pattern id="fracpattern">
  <arg>a</arg> <arg>b</arg> <arg>c</arg> <arg>d</arg>
  <expression encoding="infix">a / b + c / d</expression>
</pattern>
```

and we can use it to generate examples where the parameters are between -50 and 50 but nonzero, say. See Figure 4 shows the user view of a generated example. The user may click on links in the example to obtain more information.



Example: $\frac{1}{6} + \frac{3}{4}$

Step 1: $\gcd(6, 4) = 2$

Step 2: $\text{lcm}(6, 4) = \frac{4 \times 6}{2} = 12$

Step 3: $\frac{1}{6} = \frac{2}{12}$ and $\frac{3}{4} = \frac{9}{12}$

Step 4: $\frac{1}{6} + \frac{3}{4} = \frac{2+9}{12} = \frac{11}{12}$

Figure 4: Examples Generate by WME

Declaration Elements

Declaration elements provide information to the MeML Interpreter and control its behavior. For example, the `<internal>` element encloses code so no output is included in the resulting page. You can also use such declarations to set environment variables meaningful for the MeML interpreter. These elements will be defined precisely as we develop a prototype for the MeML interpreter.

5 MeML Coding Examples

Let's illustrate how to use MeML tags with some examples.

Example 1: The MeML code

```
<computation URL="iamc://www.SymbolicNet.org/super.wme"
             operation="sum">
  <math param=expression> <apply>
    <power/><ci>X</ci><cn type="integer">2</cn>
  </apply> </math>
  <ci param=var>X</ci>
  <cn type="integer" param=from>1</cn>
  <cn type="integer" param=to>10</cn>
</computation>
```

produces an expanded sum in the resulting page by invoking the WME server at the given URL with the specified operation and arguments.

Example 2:

```
<interaction input=x input=y action="url">
  {
    structural element,
    computation element,
    or
    verification element
  }
  that uses identifiers x and y in the expressions
</interaction>
```

The interaction element produces client-side codes to read user-supplied data and bind them to the input parameters (See Figure 5). These values together with any expressions involving the parameters are sent to the URL-identified WME server. Results can be displayed in-place or otherwise.

Example 3: With MeML, an author can define expressions, make substitutions, submit expressions to WME servers for computing support, and embed 2D and 3D plots of mathematical expressions dynamically. With examples 3 and 4, we give a complete MeML page. That teaches the *sin* function with interactive curve plotting.

We first give some internal definitions that do not result in any output in the resulting page.

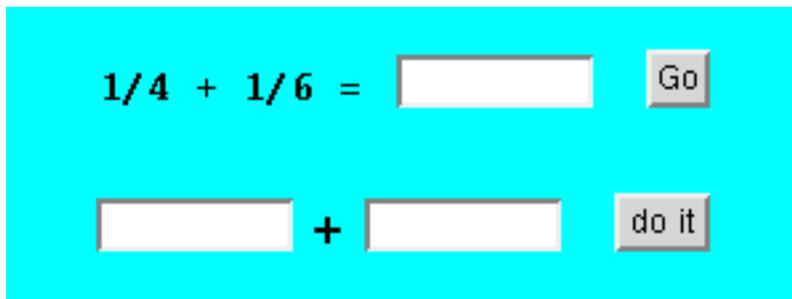


Figure 5: Generated User Interactions

```

<internal>
<variable id="vx"> x </variable>
<expression id="exp1" encoding="infix"> sin(x) </expression>
<expression id="exp2" encoding="infix"> x + 2 * PI</expression>
<substitution id="exp3">
  <idref target="exp2"/> <idref target="vx"/>
  <idref target="exp1"/>
</substitution>
</internal>

```

Then, a lesson can begin with something like:

```

<lesson title="The Sine Function" id="sinfunc">
<p>The function <em>sin</em> is important in trigonometry.
Figure 1 shows the <em>sin</em> curve. Notice that
<idref target="exp1" /> is periodic (Figure 1)
</p>

```

Figure 1 for a *sin* curve is generated dynamically with the mathgraph element:

```

<mathgraph title="Figure 1" type="2D" id="sincurve"
  URL="iamc://www.SymbolicNet.org/super.wme">
  <idref target="exp1" param="function" />
  <idref target="vx" param="variable" />
  <math id="plot-range" param="range">
    <interval>
      <cn type="real"> -10.0 </cn>
      <cn type="real"> 10.0 </cn>
    </interval>
  </math>
</mathgraph>

```

The lesson continues:

```

<p> with a period of <expression encoding="infix"
  id="p2"> 2 * PI </expression> </p>.

```

The education material here can be followed by an interactive feature that encourages student learning in the next example.

Example 4: A student can enter functions to obtain curves interactively to experiment with the $\sin(x)$ function, for instance.

```
<p> Perform your own experiments. Give an expression such as
<em>sin(x+PI)</em>, <em>sin(x-PI/2)</em>, and so on and see
the resulting curve.</p>
<guiaction id="practice">
<p> Your expression: </p>
  <expression id="exp4" encoding="infix" mode="userinput"
    action="practice"> </expression>
  <mathgraph title="curve" type="2D"
    URL="iamc://www.SymbolicNet.org/super.wme">
    <idref target="exp4" param="function" />
    <idref target="vx" param="variable" />
    <idref target="plot-range" param="range" />
  </mathgraph>
</guiaction>
</lesson>
```

Note we also reused the `plot-range`.

The Web page generated by example 3 and 4 is shown in Figure 6.

6 WME Client-Side Support

Client-side support provides for displaying pages, rendering and editing mathematical expressions, plotting curves and surfaces, manipulating geometric shapes and objects, sending computational or educational requests to the back-end, and rendering results obtained from such requests.

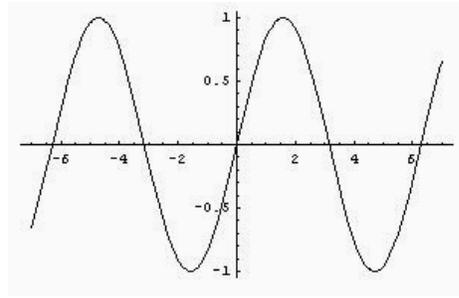
We envision to combine CSS, Javascript, DOM, Java applet, browser plug-ins, and MathML-enabled Web browsers to achieve the front-end supports.

The MeML Interpreter translates WME pages marked by the MeML tags into HTML pages supported by common Web browsers (*front-end objects*). For example, the generated mathematical expressions can be presented in MathML tags. The interaction interfaces for the end users can be HTML forms and JavaScript procedures. The results from the interactions can be rendered in-place through DHTML techniques. Thus, WME end-users can access the WME pages with regular browsers. Figure 3 illustrates the process requesting a MeML page.

An MeML Interpreter, commanded by the MeML tags, not only translates a WME page, but also generates new document objects such as texts, graphics, forms, JavaScript procedures, event triggers, and Java applets. Then it pieces them together to transform a WME page into a regular Web page. The libraries of the *MeML gadget templates* contain sets of predefined templates for easy code translation.

When combined in the translated page these templates form an *Interaction Handler* to support dynamic interactions on the client side. Figure 7 shows the role of the Interaction Handler and other possible objects in the delivered HTML page.

The function \sin is important in trigonometry. Figure 1 shows the \sin curve. Notice that $\sin(x)$ is periodic. Look at the $\sin(x)$ curve (Figure 1)



and you see the function has a period of 2π

Perform your own experiments. Give an expression such as $\sin(x+\pi)$, $\sin(x+\pi/2)$, and so on and see the resulting curve.

Your expression:

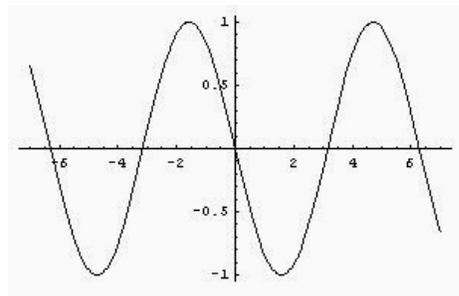


Figure 6: Teaching the Sine Function

A MathML Processor provides in-place editing and rendering functions for MathML contents. *TechExplore*, *WebEQ* or other available software can be used as the MathML processor. The processor may be a Java Applet or a browser Plug-in. The Interaction Handler is responsible for delivering MathML contents to and from the MathML processor. Our XMEC [7] (extensible mathematical encoding converter) can be used to convert infix notations to MathML before sending to the MathML processor.

The `MathGraph` element is used to generate curves and surfaces for mathematical functions. Several collaborators at Lanzhou University (PRC) are working on `MathGraph` which will also be controlled through the Interaction Handler.

7 WME Server-Side Support

Computational or educational functions in WME pages are supported by back-end resources identified by URLs. Hence any server-side programming technology can be used to provide support.

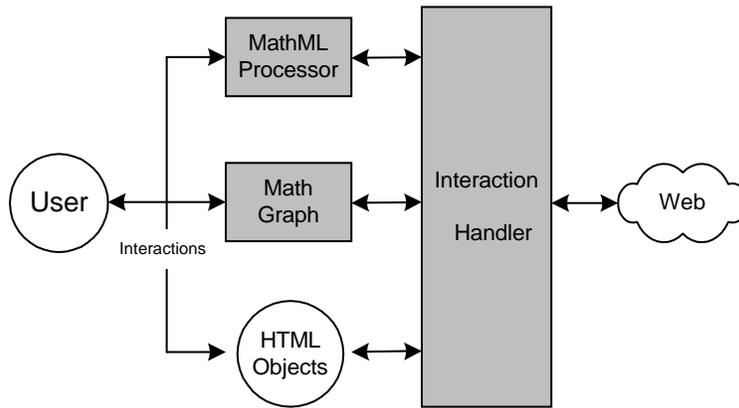


Figure 7: Client-Side Support

It is important for server-side programs to easily access mathematical computation capabilities. We plan to develop a light-weight MCP [3] client library in Java and a WME Service Broker that enables server-side programs to easily access IAMC Framework servers for a variety of mathematical computations.

The MeML Interpreter processes MeML page requests following these steps:

1. Retrieving a local .meml file according to the page request.
2. Parsing the MeML page based on our MeML DTD (XML document type definition).
3. Translating MeML elements into regular Web elements with the help of the MeML gadget templates.
4. Invoking WME servers as needed and inserting the returned content into the output document.
5. Assembling the Web page, integrating the Interaction Handler, and producing the final output.

As shown in Figure 3, a WME server request may be initiated by the MeML Interpreter (server-side interaction) or by an Internet client (client-side interaction). So, it is good to use a server side broker to manage different kinds of interactions. The WME service broker is implemented as a Java Servlet. Figure 8 shows the way the framework delivers server-side support. The WME service broker is a helper process for the WME Interpreter to reduce its work load and to handle URLs according to their schemes and protocols.

By using a WME Service Broker, we have these advantages:

- Both server-side and client-side interaction share one uniform interacting interface.
- The Service Broker can utilize the full power of Java to handle all kinds of network protocols. Hence it greatly decreases the burden on the MeML Interpreter and the front-end.

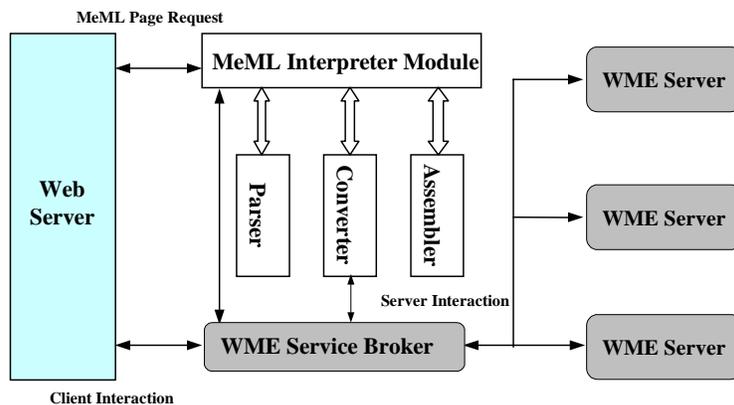


Figure 8: MeML Interpreter with WME Service Broker

- The Service Broker can perform chores such as session management and information security.

8 The Open WME Interface

To bring a multitude of useful mathematics education services online within the WME framework, we still need a uniform way to access independently developed/deployed WME servers. The Open WME Interface (OWI) addresses this need.

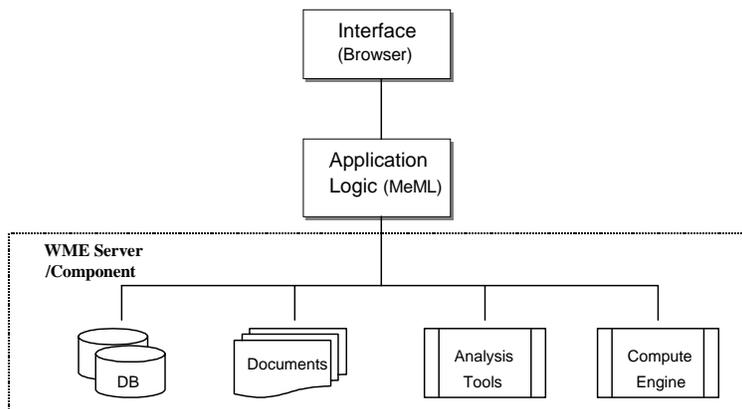


Figure 9: 3-Tier Architecture of WME Framework

OWI defines how an MeML Interpreter negotiates with a WME Server. The OWI is not an application programming interface but specifies

- Message formats between the MeML Interpreter and WME Server
- Interaction sequences or protocols according to message contents

The network connection between an MeML Interpreter and a WME Server is determined by the scheme in the server's URL. For instance, MCP can be used to access mathematical compute

engines. Other WME servers may be implemented simply as CGI programs, servlets, or server pages. We are working on the detailed specification of OWI.

9 Conclusions and Further Work

The WME framework designs a distributed system to enable Web-based mathematics education. WME is open because it uses standard Web/Internet technologies. WME is flexible because it allows educators to create teaching materials easily. WME is widely available because it is accessed by standard Web browsers. WME is powerful because it allows anyone to build WME servers that are immediately usable within any MeML page. Finally, WME can elevate the quality of the mathematics education because students and teachers have access to servers and pages built by experts world-wide, and because mathematical ideas can be reinforced with realistic examples, interactive experiments, and instant visualizations. WME has the potential of becoming a valuable tool for teachers and students of mathematics and can be applied flexibly in many educational contexts.

We have presented the initial design and architecture for WME. Much work is ahead before WME can be a successful reality. Refinements for the design, interface definitions, specification of the MeML language and its elements must be considered. Prototypes for WME servers and the MeML interpreter must be built, tested, and critically evaluated by mathematics education experts.

Further down the line, working WME prototypes should be built and field tested.

References

- [1] L. Beccerra, O. Sirisaengtaksin, and B. Waller, "On Categories of Interactive Computational Web Tools," ATCM 2000, Proceedings of the Fifth Asian Technologies Conference in Mathematics, Chiang Mai, Thailand, December 2000.
- [2] S. Linton and A. Solomon, "GAP, OpenMath, and MCP", Proceedings, IAMC'99 Workshop, July 1999,
- [3] P. Wang, S. Gray, N. Kajiler, D. Lin, W. Liao, X. Zou. "IAMC Architecture and Prototyping: A Progress Report," Proceedings of ISSAC 2001, International Symposium on Symbolic and Algebraic Computation, pp. 337-344, July, 2001.
- [4] P. S. Wang, "Design and Protocol for Internet Accessible Mathematical Computation", Proceedings, ISSAC'99, ACM Press, pp. 291-298, 1999.
- [5] A. Weber and W. Küchlin, "A Framework for Internet Accessible Software Components for Scientific Computing", Proceedings, IAMC'99 Workshop, July 1999, <http://icm.mcs.kent.edu/research/iamc99proceedings.html>.
- [6] W. Wu, "Experiments with Internet Accessible Mathematical Computation", Master's Thesis, Department of Mathematics and Computer Science, May 1998, ICM/Kent technical report ICM-199805-0003.
- [7] Xiao Zou and Paul Wang, "XMEC: An Extensible Mathematical Encoding Converter", <http://icm.mcs.kent.edu/research/xmec/>

- [8] Proceedings of the IAMC 1999 and 2001 Workshops, <http://icm.mcs.kent.edu/research/iamc.html#iamcworkshop>, July 1999 and July 2001.
- [9] <http://icm.mcs.kent.edu/research/iamc/> (IAMC homepage), <http://icm.mcs.kent.edu/research/iamcproject.html> (IAMC project homepage).
- [10] MathML International Conference 2000, www.mathmlconference.org, UIUC Illinois USA, Oct. 20-21, 2000.
- [11] Workshop on *The Future of Mathematical Communication* <http://www.msri.org/activities/events/9900/fmc99/>, Dec. 1999.
- [12] ActiveMath, <http://www.mathweb.org/activemath>
- [13] Calc101, <http://www.calc101.com/>
- [14] Ezmath, <http://www.w3.org/People/Raggett/EzMath/>
- [15] Hypermedia Browser techexplorer, <http://www-3.ibm.com/software/network/techexplorer/>
- [16] Institute for Computational Mathematics, demos of mathematical computation <http://icm.mcs.kent.edu/research/demo.html>
- [17] JavaMath, <http://javamath.sourceforge.net/>
- [18] LiveMath, <http://www.livemath.com/>
- [19] Maple, <http://www.maplesoft.com/>
- [20] Mathematical Markup Language, <http://www.w3.org/Math/>
- [21] MathLink, J/link, <http://www.wolfram.com/solutions/mathlink/>
- [22] MathScript, <http://www.mathscript.com/>
- [23] MathWeb, , <http://www.mathweb.org/mathweb>
- [24] Mathwright, <http://www.mathwright.com/>
- [25] W3C Amaya browser, <http://www.w3.org/Amaya/>
- [26] WebEQ, <http://www.mathtype.com/webmath/webeq/>
- [27] WebMathematica, <http://www.wolfram.com/products/webmathematica/>
- [28] WIMS, <http://wims.unice.fr/wims/>