Initial Design of A
Web-Based Mathematics Education Framework

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

May 11, 2002

Abstract

An approach for *Web-based Mathematics Education* (WME) is considered. WME aims to establish a framework to support authoring of dynamic and interactive mathematics education pages, end-user access through regular Web browsers, and supplying mathematical and educational capabilities through interoperable *WME servers*. The WME architecture builds on emerging Web standards, and includes the XML-based *Mathematics Education Markup Language* (MeML). An MeML interpreter translates MeML pages into regular Web pages while incorporating contents supplied dynamically by compliant WME servers. WME servers are interoperable by conforming to a common interface speci£cation.

1 Introduction

Web-based learning can extend the reach of education and signi£cantly broaden its impact and in¤uence. 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] de£nes 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 scienti£c and mathematical expressions coded in TeX/LaTeX. Some MathML is also supported. Techexplorer also allows a user to send expressions to a £xed compute server for evaluation. *MathType* [22], from Design Science Inc., supports interactive creation of mathematical notations for web pages and documents. The same company also offers *WebEQ* and *MathPlayer* [22] that 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 [26]. 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 £nd many websites providing courses and tools for mathematics education. Such sites include WIMS [28], Livemath [18], Mathwright [25], WebMathematica [27], Calc101 [13], AcitveMath [12], Maple [19], and MathWeb [24]. 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 de£ned set of topics. Authoring educational content within the WIMS scope is awkward and outside the scope is not supported. The most serious ¤aw 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 *Mathematics Education Markup Language* (MeML) and its support are considered in some detail.

2 WME Goals

The goal of our research is to produce an infrastructure, called the WME framework, for mathematics education that allows easy and systematic development of

1. mathematics educational content

2. mathematics education support capabilities

both accessible and interoperable on the Web/Internet.

With the WME framework, educators can produce MeML pages to teach speci£c lessons and topics. MeML pages can invoke support capabilities such as example generation, answer checking, graph plotting, and interactive exploration for various mathematical subjects.

Thus, content pages and support servers can be developed independently but can interoperate from anywhere on the Web. With careful design and implementation, the WME framework can become the infrastructure on which to build effective mathematics education systems on the Web. Figure 1 shows the WME objective.

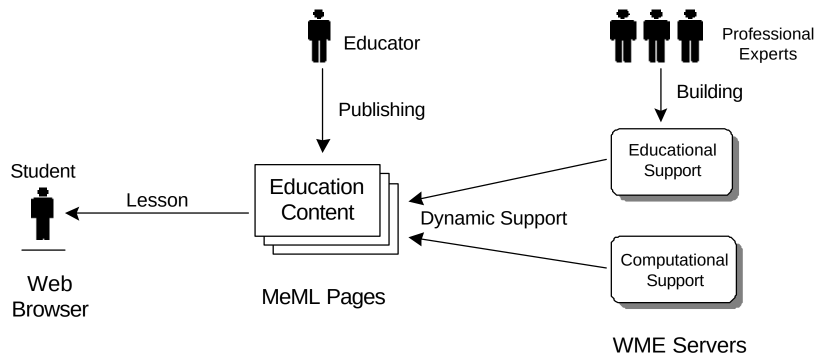

Figure 1: WME Goals

The WME framework has these characteristics:

- WME is distributed allowing everyone to put WME pages on the Web and to supply support capabilities.

- WME combines existing technologies and emerging Web standards.

- MeML pages are easy to create and simple to edit. They produce regular pages for common Web browsers.

- An MeML page can combine static content, dynamically generated educational and mathematical results, and interactions with the end user that can also be produced dynamically.

- MeML pages can invoke mathematics computation and education capabilities supplied by any WME servers identi£ed 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 2 shows the WME framework architecture.

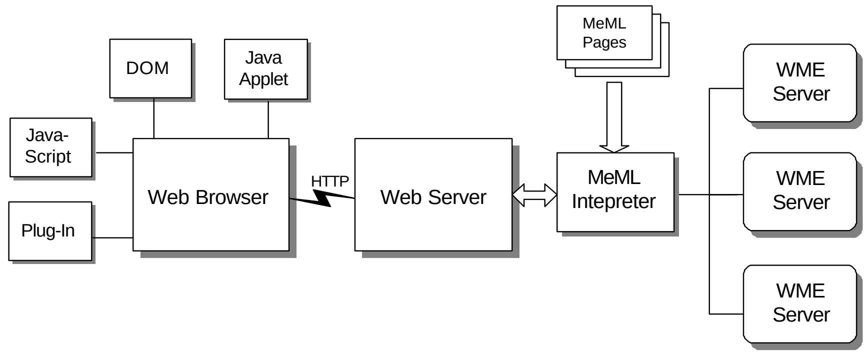

Figure 2: WME Architecture

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

- *Content-markup support*—Mathematics education contents are represented by *MeML pages* (Section 4). Basically MeML provides speci£c useful elements de£ned by *MeML tags* to be used together with HTML and MathML elements for easy content markup. MeML includes elements designed to invoke *WME servers* that supply various mathematics education capabilities.

- *Back-end support*—On the server-side, MeML pages are processed by an *MeML interpreter* (implemented as a Web server module) that dynamically transforms each MeML page, on access, into a regular Web document suitable for standard browsers. The interpreter invokes WME servers as speci£ed by MeML tags to obtain *embedable content* for building the Web page delivered to the client. Content from a WME server may contain MeML which will be processed accordingly. WME servers conform to a common interface and are accessible by URL.

  The resulting page has appropriate client-side programming (CSS, Javascript, DOM/DHTML, Java Applet, etc.) to support dynamic and interactive learning. Thus, the MeML interpreter and 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 Mathematics Education Markup Language (MeML), educational content pages written in MeML, the MeML interpreter, WME servers, and standard Web browsers.

Usage Paradigm

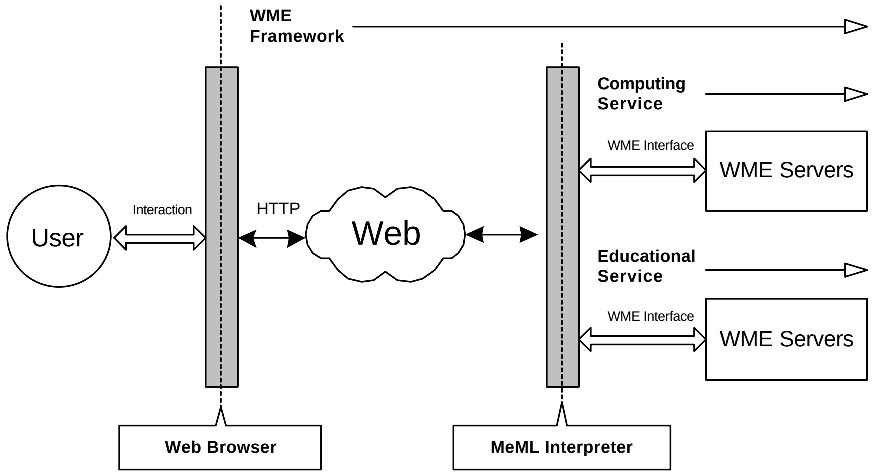

Figure 3: WME Conceptual Model

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

With WME, a new paradigm of Web-supported mathematics 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 and paces. Instructional materials may come largely from courses in schools, but the materials can be supplemented by pages from other authors globally.

- 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 a well-de£ned 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 a common interface, they can be used in any education content pages. We also envision a variety of back-end educational services including example (counter example) generation, answer veri£cation, intermediate steps production, plotting/graphing, illustration generation, word problem generation, test generation, test grading, and performance evaluation.

- Educators author interactive online courseware as a set of educational pages. Such pages are written in MeML either directly or via authoring tools that can be developed. 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 MeML pages anytime from anywhere.
- Server developers can upgrade and improve WME servers without affecting WME pages as long as upward compatibility is maintained are returned.

The *MeML Interpreter* translates WME pages into regular Web pages on-the-¤y, accessing WME servers speci£ed by MeML elements for dynamically computed content.

Figure 3 shows the WME Framework concept and the overall architecture.

4 Mathematics Education Markup Language: MeML

To support page authoring, we are developing the *Mathematics Education Markup Language* (MeML) de£ned in 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 of use,

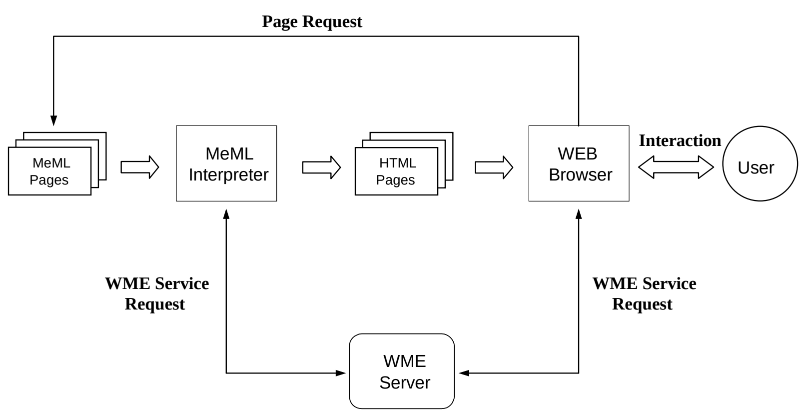

Figure 4: Accessing MeML pages

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

With MeML, an author can write HTML-like code to create mathematics education pages. An MeML page can have both £xed 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 in£x (for convenience).

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

Basically, there are £ve types of MeML elements (tags): *content elements*, *education elements*, *structure elements*, *computation elements* and *auxiliary elements*.

- Content elements—for units of knowledge such as <skill>, <terminology>, <theorem>, and <concept>.
- Education elements—for educational purposes such as, <example>, <test>, <exercise>, <diagram>, <assessment>, <diagnosis>, and <remediation>. Contents for these element can be supplied manually or can be generated. Figure 5 shows a generated example for fraction addition. The example contains links to further details.
- Structural elements—for structures in educational materials such as <lesson-opener>, <homework>, <abstract>, <summary>, etc.
- Computation elements—for explicitly invoking WME servers to obtain dynamically generated content such as <computation>, <mathgraph>,
- Auxiliary elements—for administrative aspects such as *syllabus*, *roster*, and *guide*.

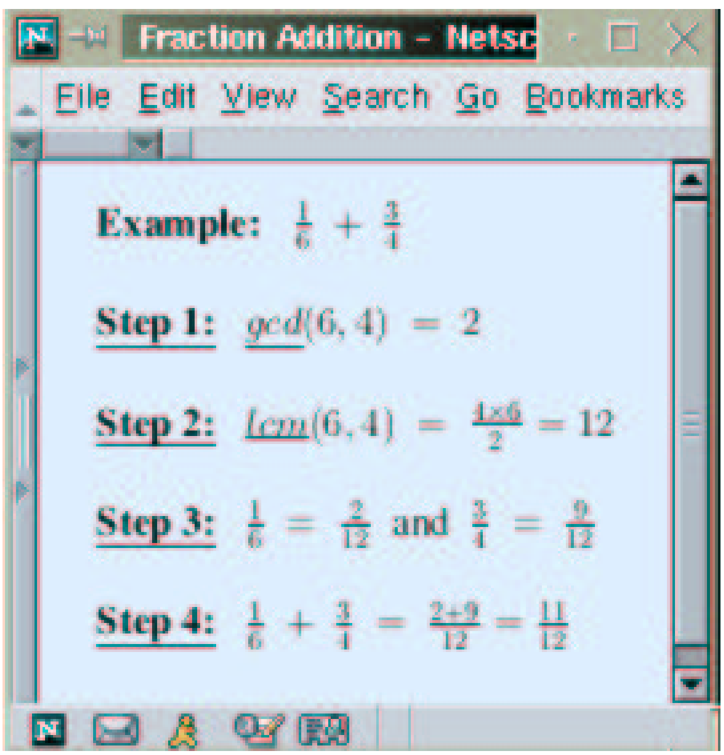

Figure 5: Generated Example

Example 1: With MeML, an author can de£ne expressions, make substitutions, generate examples, access WME servers to generate content, and embed 2D and 3D plots of mathematical expressions dynamically. With examples 1 and 2, we give a complete MeML page that teaches the sin function with interactive curve plotting.

We £rst give some internal de£nitions that are hidden from the end user:

```
<internal>
<variable id="v1"> x </variable>
<expression id="exp1" encoding="infix"> sin(x) </expression>
</substitution>
</internal>
```

The symbol `x` is a variable with `id=v1`. The expression `exp1` is given with in£x notation.

A lesson can begin with something like:

```
<lesson title="The Sine Function" id="sinfun">
<p>The function <em>sin</em> is important in trignometry.
<a href="#sincurve">Diagram 1</a> shows the <em>sin</em> curve.
Notice that <idref target="exp1" /> is periodic with a
period of <expression encoding="infix" id="p2">
2 * PI</expression></p>
```

Diagram 1 for a sin curve is generated dynamically with the `mathgraph` element:

```
<diagram title="Diagram 1" id="sincurve">
<mathgraph URL="http://icm.mcs.kent.edu/plot.wme">
  <param name="operation">plot2d</param>
  <param name="function"><idref target="exp1" /></param>
  <param name="variable"><idref target="v1" /></param>
  <param name="range">
    <math id="plot-range">
      <interval>
        <cn type="real"> -10.0 </cn>
        <cn type="real"> 10.0 </cn>
      </interval>
    </math>
  </param>
</mathgraph></diagram>
Now, <a href="#practice">experiment with sine curves</a> to
gain more insight.
```

The `mathgraph` code sends the `operation`, `function`, `variable`, and `range` values to the given `plot` server to obtain the plot.

The preceding education material is followed by an interactive exercise that encourages student learning (next example).

Example 2: A student can plot curves interactively to experiment with, for instance, the *sin* function:

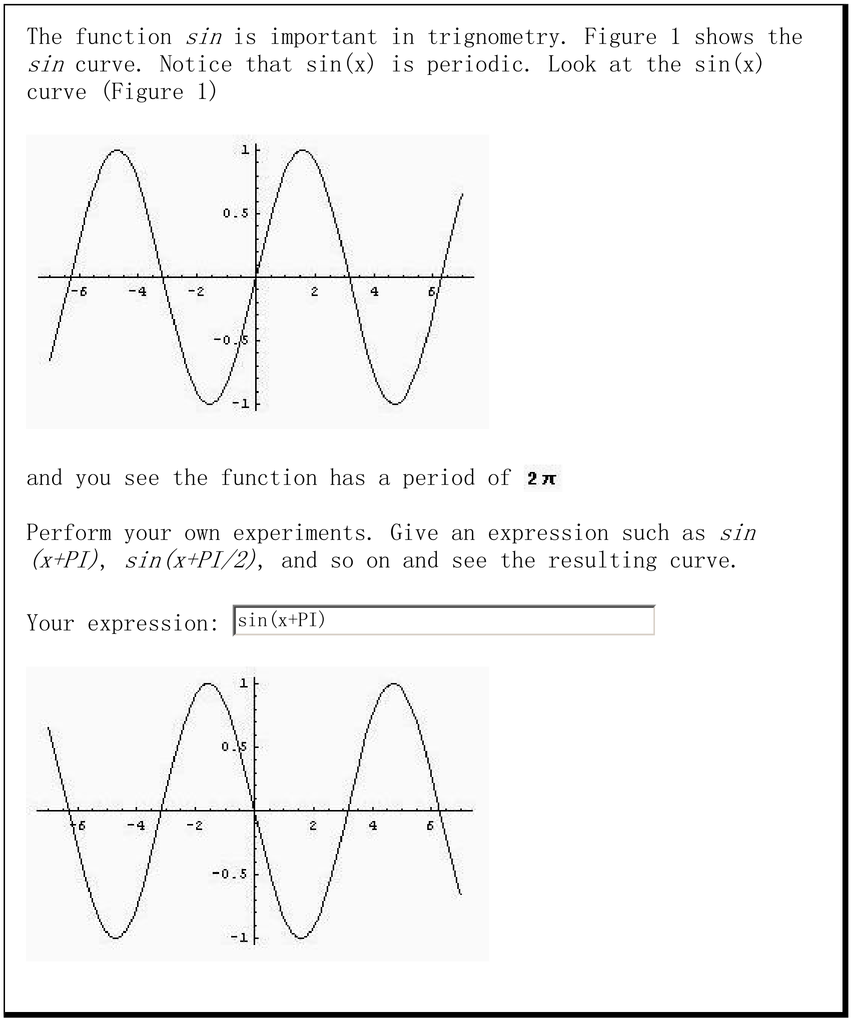

Figure 6: Teaching the Sine Function

```
<exercise title="curve" id="practice" type="interactive">
<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>
<p> Your expression: </p>
  <expression mode="userinput" id="exp2" encoding="infix"
    action="practice"> </expression>
  <mathgraph URL="http://icm.mcs.kent.edu/plot.wme">
    <param name="operation">plot2d</param>
    <param name="function"><idref target="exp2" /></param>
    <param name="variable"><idref target="vx"  /></param>
    <param name="range"><idref target="plot-range" /></param>
  </mathgraph></exercise>
</lesson>
```

The attribute `type=interactive` requests front-end supppot for interactions between the user and the WME system. Note we also reused the `plot-range`.

The Web page generated by example 1 and 2 is shown in Figure 6.

5 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 (Figure 7).

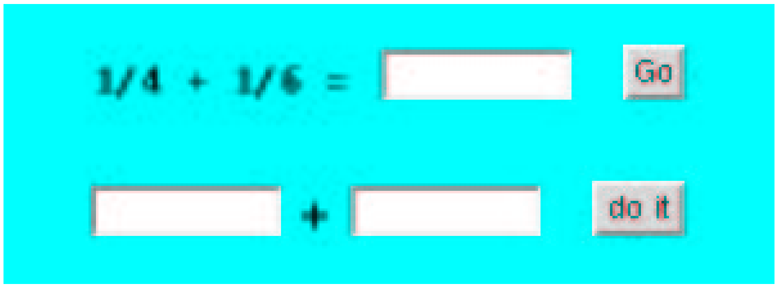

Figure 7: User Interactions

We envision a combination of CSS, Javascript, DOM, Java applet, browser plug-ins, and MathML-enabled Web browsers to provide front-end support.

The MeML Interpreter translates MeML pages into HTML pages supported by common Web browsers (*front-end objects*). For example, the generated mathematical expressions can be presented in MathML tags. The interface for end users can use HTML forms and JavaScript procedures. The results from the interactions can be rendered in-place through DHTML techniques.

In processing an MeML page, an MeML Interpreter can generate new document objects such as texts, graphics, forms, JavaScript procedures, event triggers, and Java applets. The interpreter combines such dynamically generated contents with static content in the page to create a regular Web page.

The generated page also provides an *interaction handler* (IH) to support dynamic interactions on the client side. Figure 8 shows the role of the IH and other possible objects in the page delivered to the Web browser.

Display of mathematical expression can be supported by tools such as WebEQ, MathPlayer [], and Amaya. Mathematical input from the end user can be collected in in£x form and converted to MathML by Our XMEC [7] (extensible mathematical encoding converter).

6 WME Server-Side Support

WME server-side support comes from the MeML interpreter as a web server module and from WME servers anywhere on the Internet.

The MeML interpreter is at the heart of the WME framework. It supports the MeML language, integrates services from WME servers dynamically, and delivers interactive and effective mathematics education content to learners anywhere on the Web. We are developing a prototype MeML

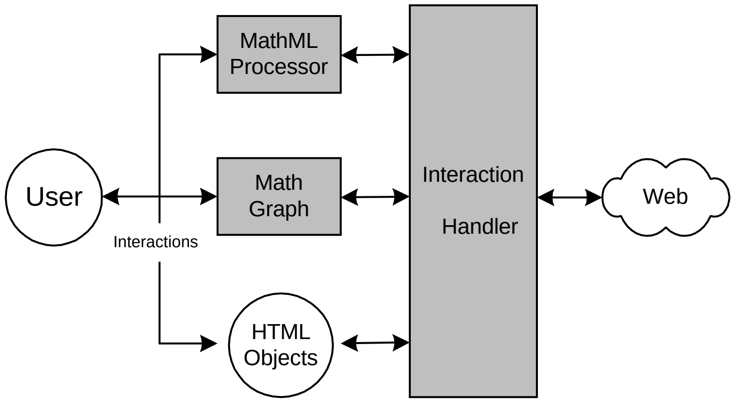

Figure 8: Client-Side Support

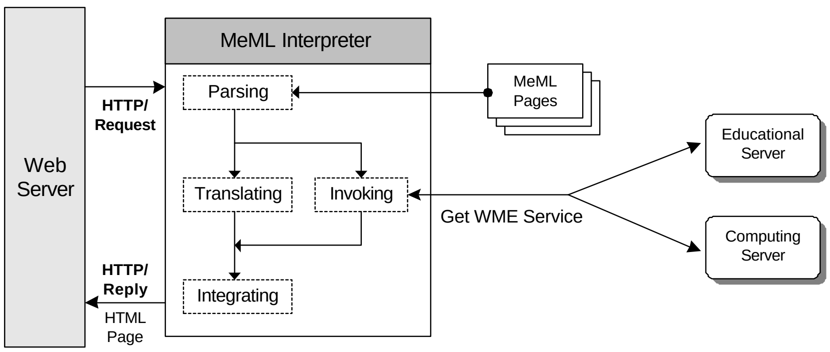

Figure 9: WME Server Side Support

interpreter as a research tool. This will help us establish a speci£cation for the functionalities and interfaces of all MeML interpreter implementations.

The MeML Interpreter processes MeML pages following these steps:

1. Retrieving as input an MeML £le from the Web server.
2. Parsing the MeML page based on the MeML DTD (XML document type de£nition).
3. Translating MeML elements into regular Web page elements.
4. Invoking WME servers as indicated in the page and inserting the content returned into the output document.
5. Assembling the output document, integrating the IH, and producing the £nal output.

Figure 9 shows the way the framework delivers server-side support.

7 Conclusions and Further Work

The WME framework designs a distributed system to enable Web-based mathematics education. WME empowers the teacher and eliminates many technical dif£culties of on-Web mathematics education. WME is open because it uses standard Web/Internet technologies. WME is ¤exible because it allows educators to easily compose and create teaching materials and obtain automatically generated contents. 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 by educators 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 ¤exibly in many educational contexts.

Speci£c features of WME not found in other mathematics educational approaches include the following:

- Educators can easily author mathematics education content for the Web with MeML, a Web publishing language specially designed for mathematics education.
- Schools can elevate the quality of mathemaics education. Well-designed mathematics education contents from anywhere can be made accessible globally 24/7.
- Experts and professionals can create support programming for mathematics education not just for a set of pages but for everyone by developing WME servers.
- Content authors can generate examples, exercises, diagrams, and other teaching materials by simple MeML tags and avoid complicated programming.
- Students can easily work with realistic examples, try alternatives, and explore theories. Bringing mathemtics education on the Web can enrich the learning experience beyond all known approaches.

We have presented the initial design and architecture for WME. Much work is ahead before WME can be a successful reality. Re£nements for the design, interface de£nitions, detailed speci£cation of the MeML language and all its elements must be considered. A uniform and effective WME server interface must be established and a way to locate and discover WME servers with the desired capabilities must be devised. Prototypes for WME servers and the MeML interpreter must be built, tested, and critically evaluated by mathematics education experts.

Further down the line, a complete working WME system must be deployed and £eld 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 Scienti£c 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] MathPlayer and WebEQ, http://www.mathtype.com/webmath/

[23] MathScript, http://www.mathscript.com/

[24] MathWeb, , http://www.mathweb.org/mathweb

[25] Mathwright, http://www.mathwright.com/

[26] W3C Amaya browser, http://www.w3.org/Amaya/

[27] WebMathematica, http://www.wolfram.com/products/webmathematica/

[28] WIMS, http://wims.unice.fr/ wims/