Mathematics Education Markup Language

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

October 17, 2002

Abstract: On-going design of *Mathematics Education Markup Language* (MeML) is presented. MeML is an XML-based language and the key component of Web-based Mathematics Education framework (WME) under investigation. MeML enables authoring of systematic and interoperable mathematics education pages that may contain *active parts* for various mathematics and educational purposes such as example generation, answer validation, difficulty diagnosis, exploration, and plotting of curves and surfaces. A server-side MeML interpreter performs the dynamic computations in an MeML page and generates a regular Web page for delivery to regular Web browsers.

1 Introduction

Web-based learning has anywhere-anytime availability and many distinct advantages. Currently, there are several problems on authoring Web pages for mathematics education: technique obstacles, computation difficulty, time consumption, lack of five ilities (modularity, reusability, exchangeability, adaptability, and machine-understandability). Recent advances in *Internet Accessible Mathematical Computation* (IAMC) [3] and MathML [21] make delivering mathematics education over Web really possible .

While various methods have been used to display mathematical formulas in Web pages and to make mathematical computations accessible via CGI programs or X Windows [17], a general and effective system for accessing, producing and delivering mathematical content is still the subject of research and development. *MathML* [21] defines an XML-based language for mathematical expressions with support for both presentation encoding (display layout) and content encoding (computation semantics). There are dozens of softwares that support rendering mathematical expressions over Web such as *Techexplorer* [16], *MathType*, *WebEQ* [27], W3C *Amaya* Web browser [26], MathML-supporting software [8, 9, 10, 12] and so on. There are also many Web Sites providing courses and tools for mathematics Education such as Mathwright [25], WebMathematica [28], Calc101 [14], AcitveMath [13], Maple [20], and MathWeb [24]. Linda Beccerra [1] gave a good summary of Web tools for interactive computation. Efforts have been made by Institute for Computational Mathematics (ICM/Kent) to build a *distributed IAMC framework* [2, 4, 5, 6, 8, 9, 12] which can support both interactive and transparent access to mathematical computation on the Internet/Web through the Mathematical Computation Protocol.

We propose MeML aiming to systematically solve the existing problems for Web-based mathematics education. MeML is a pivotal component in *Web-based Mathematics Education* (WME) [33] framework being investigated by our research group at ICM/Kent. WME works with regular browsers, makes authoring simple and easy, allows systematic access to server-side support, and enables these independently developed components seamlessly interoperable. In short we sought to create a *Web for Mathematics Education* and MeML as significant enabling technology.

2 Context of MeML

MeML functions within the WME framework that aims to support easy development and deployment of *mathematics educational content* and *mathematical & other support programming* for the Web. Content pages and support programs

*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.

can be developed independently and interoperable from anywhere over Web. Figure 1 shows the WME framework architecture.

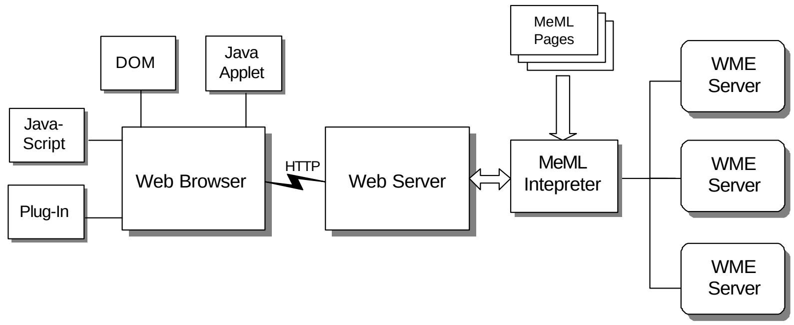

Figure 1: WME Architecture

WME is a typical three-tiered architecture and supports Web-based mathematical education through the content-markup - MeML, server-side, and client-side support. MeML, an XML-based language, is designed to author Web pages for mathematics education. An MeML interpreter provides server-side processing for MeML pages and generates dynamically composed contents delivered to regular Web browsers.

3 Goals Of MeML

MeML is designed to facilitate authoring, storing, processing, and exchanging learning materials for mathematics education over Web. MeML can represent mathematics domain knowledge, describe the actions during mathematics education (such as exercising, assessing, and etc), and do the computations for mathematics education at high school or college level through back-end computing engines. MeML is designed with following merits:

- *Easiness:* mathematics teachers can author MeML pages with any text editors or MeML authoring tools without knowledge of HTML, Web programming and etc., which overcomes the technical obstacles for them.
- *Computation:* MeML provides advanced computation with the MeML back-end support, which dramatically ease authoring tasks.
- *Efficiency:* MeML Authors only need to define *what to be taught in what ways*. How to obtain mathematics teaching stuff is internally done by MeML back-end supports.
- *Modularity & Reusability & Exchangeability:* With MeML structure elements, the content of web pages can be organized into well-defined modules, which provides reusability and exchangeability and makes full use of mathematics education resources over Web.

4 Design of MeML

Mathematics Education Markup Language (MeML)[34] is a XML based language that can embed MathML in it and can be embedded in HTML pages. MeML regards any unrecognized element as HTML elements and requires MathML tags enclosed in MeML `math` element.

With MeML, an author can write HTML-code-like mathematics education pages. An MeML page can have both fixed and dynamically generated contents. MeML interpreter controls content generation and accesses WME servers for certain requested information. One MeML page can specify all necessary interactions for Web-based Learning. All MeML elements are encapsulated in the `<meml/>` element:

```
<meml version="http://www.cs.kent.edu/ICM/MeML 1.0">
    ......
</meml>
```

MeML has five kinds of elements: *content elements*, *education elements*, *structure elements*, *computation elements* and *auxiliary elements* (not discussed here).

4.1 Content Elements

Content elements define basic vocabulary representing units of knowledge, or meta-knowledge such as `<skill>`, `<terminology>`, `<concept>`, `<expression>`, `<step>` and and etc. All content elements have following attributes: *topic*, *name*, *id*, *date*, *uri*, *encoding*, and *type*. The element `<step>` has one special attribute - *sequence* - the step order within a `<skill>` construct.

List 1

```
<skill topic="unlike unrelated fractions" name="addition"  id ="myid">
    <step sequence="1">Finding a common denominator </step>
    <step sequence="2"> Rewriting the fractions having this
          common denominator</step>
    <step sequence="3"> Adding like fractions </step>
</skill>
<theorem topic="factorization" name="unique factorization theorem">
    Any positive integer can be represented in exactly one way
    as a product of primes.
</theorem>
```

List 1 shows the skill ”addition of unlike unrelated fractions” and the theorem ”unique factorization theorem”.

4.2 Education Elements

Education elements define educational behaviors and comprise all Content and Compute elements as their children elements. Typical Education Elements are `<assess>`, `<example>`, `<exercise>`, `<question>`, `<remediation>` and and etc. Education Elements possess the same attributes as content elements. They also have special attributes. The elements `<exercise>` and `<question>` have the following extra attributes:

- *level*: defining the difficulty level. It has the values *basic*, *average*, and *advanced*. The default value is *average*.
- *number*: defining the number of exercises, examples and etc. to be generated
- *type*: defining the types of questions such as *truefalse*, *multichoice*, and *fillinblank*. The default value is *multichoice*.
- *style*: defining how users manipulate an exercise, or a question. It has the values such as *static*, *dynamic* and *generated*.

The `<exercise>` element has another special attribute *hint*, which has values such as *similar*, *formula*, *step*, *answer*, and etc. The default is *similar*. List 2 shows an example for performance evaluation on `addition of fractions`.

List 2

```
<assess name="addition of fractions">
    <terminology name="Rational Number"/>
    <terminology name="LCM"/>
    <terminology name="GCD"/>
    <terminology name="numerator"/>
    <terminology name="denominator"/>
    <skill name="addition of fractions"/>
</assess>
```

List 3

```
<exercise name="addition-of-fractions-1" topic="fraction"
          date="2002-3-18" level="basic" type="multichoice"
          style="generated" number="3">
  <pattern name="frac-add">
    <parameter> <math><apply>
      <in/><bvar>a</bvar><bvar>b</bvar><bvar>c</bvar><bvar>d</bvar>
      <interval><cn>1</cn><cn>30</cn><interval> </apply></math>
    </parameter>
    <expression><math><apply><plus/><ci type="rational">a<sep/>b</ci>
    <ci type="rational">c<sep/>d</ci> </apply></math>
    </expression>
  </pattern>
</exercise>
```

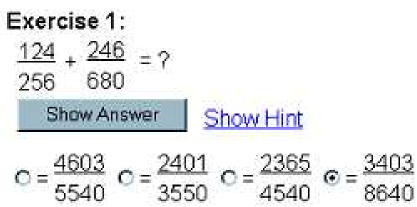

Figure 2: Exercise Sample

Exercises, homework and tests can be generated by MeML backend support. Answers submitted by students are checked to determine their understanding. WME services store students' historic performance data as the criteria for recommending further personalized learning contents. List 3 shows how `<exercise>`, `<pattern>` and MathML are used. The example says that there are a pattern named "frac-add" with four parameters `a, b, c` and `d` in the interval `[1, 30]`, and an expression "$a/b + c/d$". And it specifies generating three exercises based on the `<patter/>` "frac-add". One sample of the final HTML file is shown in Figure 2.

4.3 Structure Elements

Structure elements structure learning materials and consists of Education, and Content elements. Typical elements are `<homework>`, `<lesson>`, `<lesson-ender>`, `<lesson-opener>`, `<prerequisite>`, `<test>`, `<topic>` and etc. The `<topic>` element can be nested recursively to structure a knowledge network. They have the same attributes as Content Elements. The elements for performance evaluation such as `<homework>`, and `<test>` have a special attribute *duedate*. List 4 shows a lesson "addition of unlike unrelated fractions":—

List 4

```
<meml>
  <lesson name="addition of unlike unrelated fractions">
    <skill topic="unlike unrelated fractions" name="addition" id="myid"/>
    <example name="Addition of common fractions" number="2"/>
    <example name="Addition of unlike fractions" number = "3"/>
  </lesson>
</meml>
```

If it is defined before, it can also be retrieved from WME servers such as
`<lesson name="addition of unlike unrelated fractions" action="get"/>`.
The retrievability is one typical example of MeML exchangeability, reusability and modularity.

4.4 Computation Elements

Computation elements generate mathematical expressions, and other dynamically computed content. The *compute* and *plot* elements are examples. For instance, to obtain the factors of a polynomial, there are two ways to do it: *implicit* or *explicit*. The *explicit* is like:

```
<compute URL="wme://icm.mcs.kent.edu/wmeServer"
 operation="factor" type="polynomial">
   <variable>x</variable>
   <expression encoding="infix"> x^3-x^2+3*x-3 </expression>
</compute>
```

Authors have to specify everything such as the URL of a WME server. The *implicit* is simple such as:

```
<compute operation="factor"> <expression
encoding="infix">x^3-x^2+3*x-3</expression> </compute>
```

The element `<computation>` has only one attribute - *operation* - that has user-defined enumerate values such as `"factor"`, `"simplify"`, `"plot"` and etc.

5 Application of MeML

The goal of MeML is to shelter the authors from technical perplexity and make authoring any learning materials for mathematics education as easy as possible. MeML is designed as a sever-side technology. E.g., it requires no special browsers (Amaya, etc) or browser plug-ins (MathPlayer, WebWQ, etc) to render mathematical expressions or MathML. The working process of MeML files is almost the same as that of PHP, ASP or JSP file. MeML is a content-oriented language without presentation elements. The tool rendering MeML is MeML Interpreter - a server-side portal processor. MeML Interpreter consists of *Syntax Parser* and *Interpretation Engine*. Its working flow is Syntax parsing; accessing WME services; Converting the MeML file to one HTML file. MeML can be parsed with any XML parsers such as SAX. WME services are responsible for generating examples, homework, exercises, tests, XSLT files and etc, and returning the results to MeML Interpreter. WME interpreter engine uses the XSLT file to convert MeML files to Html files, which can be done by Apache Xalan [35]. Finally, the html files are presented to end users.

6 Conclusion and Future Work

With MeML, the problems (mentioned in Section 1) can be solved more or less. MeML lets teachers author Web pages for mathematics education easily and efficiently. Only need MeML authors care what they want without worrying about how and where the learning materials are prepared. It shelters authors from the technical complexity. All the works are based on our previous research - IAMC Framework [3]. Our solution is implemented on Windows 98/2000/xp with Apache Http server 1.34, and MathLink/Mathematica 4.1. We built an Apache module in C language, a toy MeML interpreter prototype with Xerces XML parser library by now. It is workable but far from working-well. In the future, we will focus on improving the quality and performance of the system, refining the draft of MeML specification, and implementing a full version of MeML working context - WME framework.

References

[1] L. Beccerra, O. Sirisaengtaksin, and B. Waller (2000). 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 (1999). GAP, OpenMath, and MCP, *Proceedings, IAMC'99 Workshop*, July 1999.

[3] P. Wang, S. Gray, N. Kajiler, D. Lin, W. Liao, X. Zou (2001). 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 (1999). Design and Protocol for Internet Accessible Mathematical Computation, *Proceedings, ISSAC'99*, ACM Press, pp. 291-298, 1999.

[5] A. Weber and W. Küchlin (1999), 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 (1998). 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 (2001). 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`.

[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, UIUC Illinois USA, Oct. 20-21, 2000. `http://www.mathmlconference.org`

[11] William Horton (2000). *Designing Web-Based Training - how to teach anyone anything anywhere anytime*, `John Wiley & Sons, Inc.`, 2000.

[12] Workshop on *The Future of Mathematical Communication*, Dec. 1999. `http://www.msri.org/activities/events/9900/fmc99/`

[13] ActiveMath, `http://www.mathweb.org/activemath`

[14] Calc101, `http://www.calc101.com/`

[15] Ezmath, `http://www.w3.org/People/Raggett/EzMath/`

[16] Hypermedia Browser `techexplorer`, `http://www-3.ibm.com/software/network/techexplorer/`

[17] Institute for Computational Mathematics, demos of mathematical computation `http://icm.mcs.kent.edu/research/demo.html`

[18] JavaMath, `http://javamath.sourceforge.net/`

[19] LiveMath, `http://www.livemath.com/`

[20] Maple, `http://www.maplesoft.com/`

[21] Mathematical Markup Language, `http://www.w3.org/Math/`

[22] MathLink, J/link, `http://www.wolfram.com/solutions/mathlink/`

[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] WebEQ, `http://www.mathtype.com/webmath/webeq/`

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

[29] WIMS, `http://wims.unice.fr/ wims/`

[30] `http://www.w3.org/Style/XSL/`

[31] `http://www.w3.org/XML`

[32] `http://www.wolfram.com/`

[33] Paul S. Wang, Norbert Kajler, Yi Zhou, and Xiao Zou (2002).Initial Design of A Web-Based Mathematics Education Framework, *Proceedings Internet Accessible Mathematical Computation 2002 Workshop*, Lille, France. `http://www.symbolicnet.org/conferences/iamc02/wme.pdf`

[34] MeML Document Type Definition. `http://www.mcs.kent.edu/˜yizhou/MeMLDTD.pdf`

[35] Apache XML Project. `http://xml.apache.org/`