

Math on the Web: A Status Report

January, 2001

by Robert Miner and Paul Topping

We plan on updating this report as the world of Math on the Web changes. If you would like us to let you know, visit <http://www.dessci.com/webmath> to add yourself to our Math on the Web mailing list. View this paper on-line, where the links and references are live, go to <http://www.dessci.com/webmath/status>.

Design Science
www.dessci.com
How Science Communicates™

IBM Techexplorer pages (see the comments about Techexplorer components under HTML + Components):

Advantages	Disadvantages
<ul style="list-style-type: none"> ■ Uses T_EX, a language used by legacy documents and understood by many mathematicians. ■ Doesn't require server-side support. ■ Prints with T_EX quality. 	<ul style="list-style-type: none"> ■ Requires the Techexplorer plug-in which is not free, except for its trial version which doesn't allow documents to be printed. ■ Although Techexplorer does allow inclusion of basic media formats, math can't be combined with the full range of media available in HTML pages. ■ The Web browser is merely hosting the viewer application (similar to the PDF Viewer) which must integrate its own user interface with that of the browser. This can confuse the user and limits interaction. Changes made using the browser's interface (e.g. Text Smaller or Larger) have no effect on the display.

Server-side programming (CGI, Perl, Java, ASP scripts, etc.):

Advantages	Disadvantages
<ul style="list-style-type: none"> ■ Can produce pages as in HTML + Components but avoid the image size problems. ■ Can produce pages with the widest range of browser compatibility (although this requires some challenging programming). ■ Minimal load on the client (browser) as all scripts are executed on the server. ■ Scripting needs only to target the server platform, not all the browser platforms. ■ Works well for accessing specialized software such as a computer algebra system. 	<ul style="list-style-type: none"> ■ Server-side scripting is difficult to write and debug because (a) it is running on a different computer from the browser, (b) must serve multiple clients simultaneously, (c) it runs the risk of making a production server unstable or requires setting up an identical, parallel server for testing. ■ Unlike other solutions, all load is concentrated on the server. ■ User interaction requires a round-trip to the server making interaction sluggish and unpredictable.

HTML + Components (Design Science WebEQ, IBM Techexplorer components):

Advantages	Disadvantages
<ul style="list-style-type: none"> ■ The page is HTML-based, making it easy to integrate with other Web technologies. ■ Interaction is snappy as it occurs on the client and doesn't require server-side support. ■ Interaction is powerful as scripts can generate math to be displayed on the fly. 	<ul style="list-style-type: none"> ■ Interface between math rendering component and page requires that the width and height of each equation be known in advance. This is difficult to do if the math is generated by scripting. ■ Math pages containing hundreds of individual equations can cause performance problems due to the inability of browsers to handle that many component instances. This issue can be minimized by using components in conjunction with GIFs for non-interactive equations.

Standards are paving the way to better solutions

Though there are many technologies available for Math on the Web projects, none of them is ideal. Many who consider Math on the Web projects for the first time are surprised that there is no direct support for math in HTML. After all, what is math but fancy text? In fact, this has been proposed and attempts to get it to happen are part of a history almost as long as that of the Web in general.

Efforts to get math into HTML ultimately evolved into the creation of the Math Working Group at the World Wide Web Consortium (W3C), the body responsible for Web standards. The efforts of the Math Working Group ultimately led to the development of MathML, a powerful new language for encoding math expressions for the Web. MathML is now part of an extremely powerful, standardized component-based architecture for the Web, capable of accommodating the requirements of math communication.

The HTML Platform

Each of the many technologies for putting math on the Web has its merits. However, in the long term there can be little doubt that, taken together, the collection of HTML-centric technologies that have taken shape through years of industry collaboration at the W3C represent the future of the Web. We're calling this collection of Web standards the *HTML Platform*.

The problem with current technologies for putting math on the Web is that they are too complicated and ad hoc. It often takes expert knowledge and programming skill to get different math software packages to work together, but no single approach provides a complete answer. The remedy is obviously standards. And although they take time to evolve (through organizations like W3C), once they take hold, their benefits are enormous.

As a consequence, the HTML Platform is the safest bet in the long run for lowest cost of authoring, highest level of capability, reduced vendor lock-in, and the most secure future. To get a better idea of what the HTML Platform consists of, we now take a look at the individual standards of which it is comprised.

For the question, "What language are Web pages written in?", the short answer is, "HTML, of course!". The long answer is, well, longer. Although very simple Web pages can be written in HTML alone, most pages are constructed from a conglomeration of computer languages and mechanisms working together. They go by acronyms like HTML, CSS, XSL, DOM, and MathML. These languages are defined by standards published by the W3C. (They actually call them "recommendations" rather than "standards" but they work the same way.)

Here is a list of the standards that Math on the Web developers should be concerned with, along with the role each plays in The HTML Platform:

Standard	Long Name	Role
HTML ¹³	HyperText Markup Language	Defines the document structure and acts as the glue that holds all the other pieces together.
CSS ¹⁴	Cascading Style Sheets	Associates formatting information (font, size, style, color, etc.) with HTML structural elements (headings, paragraphs, tables, etc.).
XML ¹⁵	Extensible Markup Language	The universal format for structured documents and data on the Web. Based on the earlier, more complex, SGML standard. XML is not itself a language but a meta-language used to create other languages, such as XSL, XHTML, MathML, ChemML, etc.
XSL ¹⁶	Extensible Stylesheet Language	Consists of two parts: <ul style="list-style-type: none"> ■ XSL Transformations¹⁷ (XSLT): a language for transforming XML documents ■ An XML vocabulary for specifying formatting semantics (XSL Formatting Objects)

Continued next page

Continued from page 4

Standard	Long Name	Role
DOM ¹⁸	Document Object Model	The Document Object Model is a platform-and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents.
JavaScript		Simple scripting language used to make Web pages react to user input. Typically, scripts use the DOM to modify the Web page within the browser. All execution occurs on the "client" (Web browser) as opposed to the "server" (Web server).
DHTML	Dynamic HTML	HTML plus the ability to include scripts within a Web page to accept user input, perform computations, and modify the Web page within the browser using the DOM interface in order to effect user feedback.
XHTML ¹³	Extensible HTML	An XML-based version of HTML.
MathML ¹⁹	Mathematical Markup Language	An XML-based language for describing mathematics for both presentation and computation.
CML ²⁰ and other MLs	Chemistry Markup Language, etc.	Other domain-specific markup languages may be used in a Web page to describe structured text and/or data.
Applets and Plug-ins		Separately downloadable mini-applications that extend the capabilities of the browser. These may be written in Java for cross-platform compatibility or in C++ for performance. The WebEQ™ Equation Input Control is a good example, providing a way for users to enter math notation and send it to the Web page as MathML.

You can find out more about most of these pieces of the HTML Platform by visiting the W3C's Web site at <http://www.w3.org/>. Links to specific standards are also listed in References.

All modern Web browser makers have committed to supporting the standards of the HTML Platform. For the Math on the Web author, it is largely a matter of when, not if, a browser implements a particular HTML Platform standard. Like all new technology, the HTML Platform is not standing still. The W3C and its member organizations are constantly pushing the HTML Platform forward. Although the Math on the Web author must still be concerned with questions about whether *this* technology is supported in *that* browser, this is inevitably the case with leading-edge technologies.

The role of MathML in a dynamic math page

MathML support is already widespread enough that it has begun to play a pivotal role for *dynamic math* in Web pages. Part of the appeal of the Web as a

medium is that it is interactive. Information is presented in an active environment with many kinds of interactive controls. In the case of math content, MathML-capable components can be "wired together" to create dynamic math and science pages.

The most basic situation are MathML "islands" in the Web page, which display as traditional mathematical notation. In some browsers like Amaya and Mozilla, display of MathML islands is now built-in. With Netscape and Internet Explorer, MathML can be rendered via plug-ins or applets.

The role of MathML islands in an HTML document is similar to the role of T_EX's math mode within a T_EX document. However, MathML has the potential to go much farther than T_EX's static math. MathML equations can be manipulated using JavaScript, and styled using CSS. Equations can be hooked up to HTML controls like buttons and pulldown menus. Applets and plug-ins can send MathML-encoded equations to server-side software such as computer algebra systems or scientific visualization software.

The combination of MathML and scripting facilitate an enormous range of dynamic math applications. Here are some possibilities:

- When the user clicks on a button in the page, a script is invoked that modifies a MathML island using DOM access. The user sees the math change based on the click. A typical use might be to display a hint to a student in a lesson.
- An interactive assessment page allows the student to invoke an equation editor applet. When the user closes the editor window, the MathML for the equation is sent as input to a script that sends it back to the Web server. Software on the Web server checks the student's answer and sends the result back to the script running in the page which informs the user of his/her success/failure.
- A Web page that works as a special-purpose calculator and grapher allows the user to make radio-button and checkbox choices and enter numerical coefficients. A script invoked by a "Compute" button generates MathML that will be sent to a Computer Algebra System or other computational component running on the client or server computer. The results are shown to the user both as a graph (using a graphing applet) and in standard mathematical notation.

Much as HTML is the glue that binds the various parts of the Web page together, MathML is the glue that binds the mathematical and computational elements together. MathML is the language with which the equation editor applet, the Computer Algebra System, and the graphing applet communicate.

The HTML Platform vs. proprietary, single-vendor platforms

The biggest choice facing the Math on the Web author is whether to go with a proprietary, single-vendor platform, or to stick with an open standards-based, multi-vendor platform. The standards-based platform is the better choice in the long run, but the case isn't so clear when you have to do something today. This is a very common dilemma for technology adopters. The proprietary, single-vendor solution has some advantages, especially in the short term:

- Available sooner;
- Easier to evaluate for possible adoption;
- Once adopted, easier to learn as there is a single source for all documentation, knowledge, and support.

The proprietary, single-vendor solution has some big disadvantages which grow as time passes:

- The vendor is slow to implement new technologies as it has to do all the work itself.
- The customer gets locked in to the proprietary solution as there are no alternate vendors.
- Switching to the open standards-based, multi-vendor solution can be difficult and costly as the customer has a huge investment in training, expertise, and experience in dealing with the single vendor's system, much of which may have to be discarded.
- It is usually more difficult for a proprietary system to interoperate with standards-based systems. This is what occurs when Adobe Acrobat, IBM Techexplorer, and other viewers take over the browser. The viewer's user interface must be overlaid on top of the browser's, thereby confusing the user.
- The proprietary vendor can charge higher prices, especially after the initial purchase (e.g. support, upgrades, etc.) because of the customer's difficulty in switching vendors.

In the long term, the open standards-based, multi-vendor solution has big advantages that grow even larger as time passes:

- Multiple, competing vendors for each component of the solution (authoring systems, in-page editors, applets, browsers, etc.) mean more customer choice of product and vendor, lower prices, more features. This is capitalism at its finest.
- There is much less vendor lock-in. Switching vendors is easier for a single part of your solution than for the whole thing.
- It will be easier to find training, support, contract help, etc. as the technologies you are dealing with will be mainstream, often not specific to Math on the Web.
- Similarly, trained and experienced personnel will be easier to find and train as the skills you will be looking for will be common.

What's available now

There are already a number of software packages that have some level of MathML support. In particular, Design Science is one of the oldest and most respected vendors of MathML software. Our WebEQ software was the first commercially available software with MathML support, MathType™ was the second.

The following list touches on some of the highlights:

From Design Science (DSI)²¹:

- MathType — DSI's premiere equation editor can export equations as MathML. MathType gives an easy way of authoring MathML expressions.
- WebEQ Math Viewer — this scriptable Java applet displays math expressions in Web pages, and can dynamically update the display in response to mouse gestures or other controls, such as buttons, in a Web page.
- WebEQ Equation Input Control — The Equation Input Control (or EIC) is a Java equation editor applet. The EIC gives readers a way to enter math on a Web page, in addition to the other standard HTML input controls like text areas and radio buttons. The EIC is configurable to meet a range of authoring needs.

From the T_EX Community²²:

- tex4ht²³ — this free Perl program is a very powerful HTML translator, generating HTML + MathML from L^AT_EX.
- PassiveT_EX²⁴ and xmltex²⁵ — these free macro packages facilitate using T_EX to directly typeset MathML.

From Wolfram Research²⁶:

- Mathematica 4.x — this powerful scientific computation platform can import, export, and evaluate MathML.

From IBM:

- IBM Techexplorer²⁷ — the IBM Techexplorer plug-in takes over the browser window to display T_EX documents by default. However, with some effort, Techexplorer can also be used as a component

within the HTML platform to display and manipulate MathML expressions in a way similar to the WebEQ Math Viewer.

Browsers:

- Amaya²⁸ — the demonstration and test-bed browser from W3C supports both display and editing of MathML.
- Mozilla²⁹ — the open source version of the Netscape browser currently implements most of MathML. There is a strong likelihood MathML support will appear in the Netscape-branded version of the browser before long.
- Internet Explorer³⁰ — Microsoft has introduced a new plug-in technology called "Behaviors"³¹ that allows much better integration into a Web page than other plug-in technologies. More on this below.

For a more comprehensive list of MathML-enabled software, consult the W3C Math home page.

What's in the works

Design Science²¹ will be introducing a number of products supporting MathML and the HTML Platform in early 2001:

MathPlayer

- MathPlayer™ is a state-of-the-art plug-in that enables Internet Explorer (IE) to display MathML. Developed in collaboration with Microsoft using IE's Behavior technology, MathPlayer seamlessly integrates MathML into Web pages, automatically matching font sizes and styles, flowing and aligning with surrounding text, and printing at high-quality. MathPlayer is designed to be part of the HTML Platform, and can be scripted with JavaScript, and styled with CSS.

MathPlayer will be available free-of-charge.

MathType 5 with MathPage technology

- The next version of Design Science's premiere MathType equation editor will revolutionize the process of exporting a Word document with math

to the Web. DSI's MathPage™ technology augments Word's "Save As HTML" feature in powerful ways.

MathPage uses the full power of the HTML platform to create HTML pages with math as MathML islands for display in MathML-capable browsers, such as IE with MathPlayer, Mozilla, and Amaya. For maximum backward compatibility, MathPage can also produce math as GIF images, with high resolution images for printing, all perfectly aligned with the surrounding text.

WebEQ 3

- WebEQ, the oldest and most comprehensive set of tools for building dynamic math Web sites, is getting a load of exciting new features in its next release. The core WebEQ MathML engine has been completely updated for MathML 2 and features better rendering, more intuitive editing, and automatic line breaking. WebEQ 3 will also introduce direct authoring of *content MathML* which encodes both how an equation should evaluate and how it should look.

The WebEQ Math Viewer and Equation Input Control will both offer many new scripting and configuration options, and the WebEQ MathML Editor and Page Wizard programs will provide Web developers with the most powerful tools available for directly authoring and processing MathML.

Conclusion

A great math site is great because of its content, not its technology, and that will never change. But the technology is important too, and the many choices between solutions, none of which are ideal, have posed a real obstacle to would-be Web authors and readers. Standards have been developed at the W3C to address the problems, and the HTML Platform that has emerged from these efforts is a very powerful and flexible architecture for math and science publishing on the Web. But as always, for authors making choices today, the paramount question is "What works now, and when's the next release?"

On that point, the current picture is quite positive. As we have seen, MathML is becoming widely supported. The First International MathML Conference³² was held this fall, and the amount and quality of MathML implementation work on display was impressive. By the end of next year, excellent MathML support will be freely available in most major browsers. Several computer algebra systems currently support MathML, and more are on the way. Excellent authoring tools are available for developers. And much, much more is on the way, from Design Science and others.

Math on the Web has come a long way, and has a bright future. Design Science believes the coming year is going to be a very big one for Math on the Web. The HTML Platform has matured over the last two years, and Math on the Web is well on the way toward claiming the benefits.

References

- ¹The Math Forum — <http://www.mathforum.com/>,
- ²Math Goodies — <http://www.mathgoodies.com>
- ³Index of Famous Curve — <http://www-groups.dcs.st-and.ac.uk/~history/Curves/Curves.html>,
- ⁴Prometheus — <http://www.prometheus.com>,
- ⁵LiveMathBoard — <http://www.livemath.com/livemathboard>,
- ⁶Wiley e-Grade — <http://jws-edcv.wiley.com/college/egrade/>,
- ⁷Knot a Braid of Links — <http://west.camel.math.ca/Kabol/index.html>
- ⁸Euclid's Elements — <http://aleph0.clarku.edu/~djjoyce/java/elements/elements.html>,
- ⁹Twin Dragon — <http://cm.bell-labs.com/who/jelena/TwinDragon/>
- ¹⁰arXiv — <http://xxx.lanl.gov>
- ¹¹MathSciNet — <http://www.ams.org/mathscinet/>
- ¹²Sloane's On-Line Encyclopedia of Integer Sequences — <http://www.research.att.com/~njas/sequences/>
- ¹³HTML and XHTML — <http://www.w3.org/MarkUp/>
- ¹⁴CSS — <http://www.w3.org/Style/CSS/>
- ¹⁵XML — <http://www.w3.org/XML/>
- ¹⁶XSL — <http://www.w3.org/Style/XSL/>
- ¹⁷XSLT — <http://www.w3.org/TR/xslt>
- ¹⁸DOM — <http://www.w3.org/DOM/>
- ¹⁹MathML — <http://www.w3.org/Math/>
- ²⁰CML — <http://www.vsms.nottingham.ac.uk/vsms/talks/chemweb.html>
- ²¹Design Science, Inc. (MathType, WebEQ, MathPlayer) — <http://www.dessci.com>
- ²²TeX (TUG: TeX Users Group) — <http://www.tug.org>
- ²³tex4ht — <http://www.cis.ohio-state.edu/~gurari/TeX4ht/mn.html>
- ²⁴PassiveTeX — <http://users.ox.ac.uk/~rahtz/passivetex/>
- ²⁵xmltex — <http://www.dcarlisle.demon.co.uk/xmltex/manual.html>
- ²⁶Wolfram Research (Mathematica) — <http://www.wolfram.com/>
- ²⁷IBM Techexplorer — <http://www-4.ibm.com/software/network/techexplorer/>
- ²⁸Amaya — <http://www.w3.org/Amaya/>
- ²⁹Mozilla — <http://www.mozilla.org>
- ³⁰Microsoft Internet Explorer — <http://www.microsoft.com/windows/IE/>
- ³¹Element Behaviors in Internet Explorer 5.5 — <http://msdn.microsoft.com/workshop/c-frame.htm?workshop/essentials/versions/IE5behave.asp>
- ³²The First International MathML Conference — <http://www.mathmlconference.org/>

Design Science
www.dessci.com
How Science Communicates™

MathType, WebEQ, MathPage, and MathPlayer are trademarks of Design Science, Inc.
All other company and product names are trademarks and/or registered trademarks of their respective owners.

Copyright ©1999-2000 by Design Science, Inc. All rights reserved.