Firefox OS Web Apps for Science

Raniere Silva and Frédéric Wang

Mozilla MathML Project

April 19, 2014

Contents

Acknowledgments

Introduction

The Web has become an integral part of our daily life. Some of the reasons for its success are its open nature and the way it enables anyone to get access to knowledge and to create projects. The Mozilla community has worked since the early days of the Web to guarantee openness, innovation and opportunity.

The Web was initially created at the CERN to share knowledge between researchers. But although it was invented by scientists, we still have not seen the same positive impact on scientific practice. There are two main reasons that could explain this situation and they are actually related each other.

The first one is a human problem. Researchers have kept teaching students to write papers for publication in journals and to avoid sharing their detailed results because of competition between academic circles. This means that most scientists ignore how to use tools to publish Web content and do not have the culture of openness and collaboration. Mozilla Science Lab was launched last year to remedy that problem and build educational resources, tools and prototypes for the research community.

The second one is more technical: the lack of tools in the first years of Web and more recently the new tools available don't be used at the early years of undergraduate and graduate courses. In science, technology, engineering and mathematics there is one extra technical problem: we still lack a cross-compatible way to publish mathematics on the Web despite the publication of the MathML standard in 1998. The Mozilla MathML Project was launched in 1999 and in a few years, the team produced a good MathML implementation in Gecko together with tools to publish mathematics on the Web. MathML finally became part of HTML5 thanks to Mozilla's effort, but other Web rendering engines still have limited support or even no support at all. This means that scientists either stay outside the Web (e.g. exchange only PDF documents) or rely on some workarounds to publish mathematical content on the Web, with their inherent limitations and issues.

In recent years, the mobile market has grown considerably and more and more people are using mobile devices to access the Web. Mozilla has been working on an open-source operating system for these mobile devices that relies on open standards and in particular Web technologies. Thanks to Gecko's good support for MathML and HTML5 in general as well as Mozilla's long experience with community submission, we now have the opportunity to build a family of Scientific Web applications for Firefox OS devices. Some of our early prototypes are presented in this paper.

In a first part, we will review the Web platforms and focus on how the technologies can be used for science. We will present the classical features as well as more recent improvements that have been integrated in Mozilla projects recently such as the Open Type MATH table, WebGL, Web Components or TeXZilla. Some of this work has been made during the crowdfunding project "Mathematics in ebooks" which has also resulted in the creation of collection of scientific documents using advanced Web technologies.

In a second part, we will study how to use these technologies to write Firefox OS Web apps for Science and introduce a few of the tools recently developed by the Mozilla MathML team.

The authors have written this paper using collaboration tools like GitHub and all the sources, programs and tools presented here are Open. Because PDF format has been requested for submission to the MathUI workshop, we had to provide our demos separately instead of integrating them directly in the document. However, we invite the reader to test the demos in a Gecko browser. A Web version of that document is also available for online reading.

Chapter 1

The Web Platform

1.1 Overview

All the technologies presented in this chapter are based on Web standards and should be supported by any Web rendering engines. We will particularly be interested in Gecko which has the best native MathML support and is the core of FirefoxOS. We will present some of the improvements that have been made by the Mozilla MathML team.

All these technologies should be usable in HTML documents. This obviously includes Web pages but also EPUB ebooks, HTML mails, Browser add-ons, or Firefox OS Web apps. For example, it is possible to receive and send emails with mathematical equations using Thunderbird or Seamonkey's mail client. In this paper, we will mainly focus on Firefox OS Web apps but keep in mind that all the features apply in other context too.

1.2 Basic HTML5 Features

The main language is HTML5, which allows to create pages with headers, paragraphs, tables, hyperlinks etc The well-known CSS language is used to apply specific style to HTML5 and is powerful enough to produce advanced designs. Finally, DOM/Javascript provides a programming language and enables interactive documents and complex user interface. New HTML5 elements gives other possibilities . For example the document `pendulum-20131125` of the "Mathematics in ebooks" project uses the the `<video>` tag to insert some sequences of a physics lecture.

For scientific documents, we need two other features: creating graphs, schemas, diagrams etc and writing mathematical formulas. For the former, simple PNG images might be enough. However, Web rendering engines also support the SVG language to let authors write scalable images using some simple drawing primitive. Many programs are available to generate scientific schemas in SVG formats. Mathematical equations can be viewed as an extension of text layout and thus requires a good integration within HTML as done by Gecko's native MathML. The document demos/1-mathml-in-html.html shows how various font properties apply to MathML text via CSS, the good alignment of inline equations and its the scalability.

One of the nice feature introduced some years ago in Gecko is the possibility to integrate MathML equation inside SVG images via the `<foreignObject>` element. People can then create scientific schemas with mathematical equations. We will also use this property later when we introduce `<canvas>`. See demos/2-mathml-in-svg.svg for an example of a SVG schema with MathML equations inside. LaTeXML 0.8 can generate such schemas from the LaTeX commands of the TikZ package.

demos/3-mathml-javascript.html is a small example of an interactive MathML formula. Javascript is used to allow the user to highlight each term of a 3-dimensional determinant and understand each term of the Sarrus' rule. Note that no particular Javascript API is needed, you just modify the MathML tree via

the standard DOM interface and the rendering is automatically updated. This example is taken from the "Mathematics in ebooks" which contains many other examples of this type.

To conclude this review, we briefly mention classical Web features like Unicode characters, complex text layout (e.g. ligatures), right-to-left directionality, copy and paste, automatic line breaking and accessibility to users with disabilities. In general, these are well handled by Gecko. The two first works well in MathML too and right-to-left MathML has also been implemented in Gecko four years ago. For the fourth one, we have proposed a MathML Copy add-on for Firefox as a temporary solution. MathML line breaking is important for small screens but to our knowledge, no CSS-compatible implementations exist. Finally, MathML accessibility is not available in Gecko contrary to other Web systems. We expect to bring improvements to these three last points in order to get a complete Web platform for mathematical user interfaces.

1.3 Styling of Mathematics

As seen in previous examples, one can use CSS the standard way to apply specific styles to mathematical equations. However, perhaps the most important style that scientists and publishers want is the rendering with a given mathematical font. Gecko has always used TeX heuristics to position scripts, fractions, roots etc but these rules do not necessarily work well for other fonts than Knuth's Computer Modern. Moreover, in order to draw stretchy and large operators the MathML code requires to pick and assembly specific glyphs from math fonts. So far, Gecko only some Unicode constructions and had few private per-font tables to do that. This means that without the appropriate fonts installed, the quality of the MathML rendering could be very low.

In order to solve that problem, we have started to implement Microsoft's OpenType MATH table, which is currently undergoing standardization at the MPEG group. We have started to use this table in order to provide a generic support for MATH fonts. In a nutshell, the main features are:

1. Some OpenType Tags to provide alternate form of glyphs. For example, we implemented the "ssty" feature to adjust the size of glyphs like primes when they are used as scripts.

2. Font parameters to define precisely the gaps, shifts, kerning etc of mathematical objects. Most of them are extensions to the TeX rules so it is easy to integrate them in our MathML rendering engine.

3. Font parameters specific to each glyph. At the moment, we only considered italic corrections.

4. A table to draw stretchy and large operators such as parenthesis, radicals or summation symbols. We have started to use that table for our operator stretching code.

There are various mathematical fonts with an OpenType MATH table available and most of them are distributed with a TeXLive distribution. Windows and Mac systems have respectively Cambria Math and STIX available. We have also experimented with PackageKit auto-installation on Linux. This means that missing appropriate fonts will become very unlikely on Desktop. We also have plans to make these fonts preinstalled on FirefoxOS. In any case, fallback downloadable Web fonts can also be used.

The page demos/4-mathml-fonts.html shows how to get various font style for a document. At the moment, it will not work correctly without a Nightly version of Gecko and some additional patches.

1.4 TeXZilla

Mathematical formulas are complex and hence writing mathematics is difficult. Some tools like WYSIWYG editors or handwriting recognition might help. One input method easy to implement is the conversion from a simple syntax like LaTeX into MathML. Even if there are tons of such converters, only a few of them are implemented in Javascript. The remaining ones try to tweak the output to workaround browser limitations, are not standalone Javascript module usable in add-ons and do work with Unicode characters or right-to-left mathematics. Hence we decided to write yet another converters to focus on Mozilla's needs.

We thus wrote TeXZilla, a LaTeX-to-MathML converter generated with the help of Jison and relies on a LALR(1) grammar and on the unicode.xml file of the XML Entity Definitions for Characters specification. It accepts arbitrary Unicode input as well as right-to-left mathematics, something that is important for the world-wide aspect of the Mozilla community and more specifically Arabic mathematics.

The public API contains a `toMathML` function to convert a LaTeX string into a MathML DOM but also a `toMathMLString` function when the DOM is not available (e.g. in nodejs). There is also a convenient `getTeXSource` function to extract the LaTeX source from the MathML output and a `toImage` function to embed the MathML output in a SVG image (see the next section).

TeXZilla can work in any CommonJS program, in Web page, as a command line program or as a Web server. We have submitted a Firefox add-on that has been approved by the reviewers. We have also already integrated it in various other Mozilla tools: in CKEditor for the Mozilla Developer Network, in the Seamonkey/Thunderbird composers or in a Firefox OS webapp (see next chapter). Finally, a prototype `<x-tex>` custom element is also available (see Web Components). We expect it will become an important library for future Mozilla applications.

1.5 Canvas and WebGL

The `<canvas>` element has been introduced in HTML5 to draw graphics via JavaScript. This element may be more convenient than SVG in situations where you want to generate schemas dynamically. For example demos/5-canvas.html shows the typical graphs associated to a RLC circuit that are automatically updated when the user changes the frequency of the current. It is also possible to use WebGL, opening the possibility to draw 3D scientific schemas. For example chemdoodle relies on WebGL to provide 3D representation of chemical structures.

As for SVG, 2D/3D scientific schemas created via the `<canvas>` might require mathematical formulas. This is possible since we saw that MathML can be inserted in SVG via a `<foreignObject>` and moreover SVG images can be inserted in both the CanvasRenderingContext2D and WebGLRenderingContext. One can for example use TeXZilla's `toImage` function to generate such SVG images dynamically. The page demos/6-mathml-in-webgl.html shows an example of a 3D schemas with mathematical formulas inside.

1.6 Web Components

One of the recent HTML5 features that is currently being implemented in Web rendering engines is Web Components. The idea is to allow Web developers to create their own custom HTML elements. These elements are made of the so-called shadow tree, which is an "internal" representation from basic elements andCSS style to which we attach some behaviors with DOM events and Javascript. This is typically useful to create some reusable widgets. Mozilla's Brick project is a collection of such elements with demos. Web Components are currently not fully implemented natively in browsers but some polyfill libraries like X-Tag allows to emulate their behavior.

Web components can obviously be very helpful to design user inteface for sciencific Web app. For example, the specific interactive drawing of demos/5-canvas.html for RLC circuits could be become a generic widget `<x-graph>` to draw scientific graphs with with some user interface to configure the curve parameters. As a proof-of-concept, we have used the X-Tag library to create a custom `<x-tex>` tag. The behavior is simple: it automatically converts its LaTeX content into MathML using TeXZilla, see demos/7-web-component.html. The Mozilla MathML team is following the developments in Web components and plans to create more custom elements for sciencific web apps, so that Web developers can reuse them.

Chapter 2

Firefox OS

2.1 Overview

Firefox OS is the open-source operating system targeting mobile devices being developed by Mozilla and based on Web technologies. All the apps in Firefox OS are Web pages and to create a app from a previous Web page you only need a manifest file that store some metadatas about the app and a figure to be the icon. The documents `demos/app/manifest.webapp` and `demos/app/icon.png` show how create a app for `demos/app/index.html`.

In the next section we give more details of what we call "Math Suite" and after that we present some of the demos wrote by Mozilla MathML community.

2.2 Math Suite

Like an office suite is a collection of softwares intended to be used by knowledge workers an Math Suite is a collection of softwares intended to be used by someone that make intensive use of math and should have tools like markup converters, WYSIWYG editor, handwriting recognition and more.

Many of the applications in an Math Suite share a core set of functions and have this functions available in a open source Javascript library will make easy to develop new applications in the same way as Basic Linear Algebra Subprograms (BLAS), a specification for common operations using scalar and vectors, wrote in 1979 are still used as a building block in higher-level math programming languages and libraries.

The authors of this work start writing this Javascript libraries and some demos that use it.

2.3 Math Cheat Sheet

This is the most basic app that someone can write since it is a collection of common K-12 equations available as a app.

The equations are organized in sections and a table of contents is provided to help jumping from one section to another. Nice features that can be add in the future include:

- link to resource like Wikipedia related to the equation,
- search bar to quickly find the desired equation,
- customization of equations, ...

2.4 TeXZilla App

This is a note block app for math that take (La)TeX as input.

Conclusion