Maps for HTML: A New Media Type and Prototype Client for Web Mapping

Peter Rushforth
Natural Resources Canada
560 Rochester St
Ottawa, ON
+1 613 759-7915
Peter.Rushforth@Canada.ca

ABSTRACT

The Web has a long history of Web mapping, being originally described at the first WWW Conference in 1994. Web maps have evolved significantly since then, and patterns have developed. The patterns of modern Web mapping underlie mapping programs across economic sectors. Key issues which remain are that Web standards do not directly address the needs of mapping, and Web mapping standards do not rely on Web architecture. As a result of this lack of collaboration, Web mapping and Web standards continue to evolve independently, with little coordination based on a broader interest. This has led to a situation where newcomers to Web mapping are faced with the problem as to what technologies to use for creating and publishing Web maps, of which the unintended consequence is increasing centralization of Web mapping.

This paper documents the results of design and development done by Natural Resources Canada within the scope of the Maps For HTML Community Group. A declarative Web map extension to the HTML standard is proposed, together with a new supporting hypermedia type. Taken together, the proposed standards will progressively support simple to advanced Web map applications, including considerations of layers, projections and feature styling. If widely implemented, the proposed Web standards could help realize the value of the substantial investments in spatial data across all sectors of society.

The paper concludes with several propositions drawn from the discussion, and proposed actions to be undertaken by the Maps for HTML Community Group, in which the reader is invited to participate.

General Terms

Human Factors, Standardization, Languages, Legal Aspects

Keywords

Web maps; Mapping; Cartography; Standards; Hypermedia

WWW 2016 Companion, April 11-15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.

http://dx.doi.org/10.1145/2872518.2890467

1. INTRODUCTION

Maps have been a part of the Web from very early on. Web maps were first described by Steve Putz of Xerox PARC, at the first WWW Conference in 1994 [1]. The modern Web is a dramatically different technological and social environment than the early Web of the 1990s. Web technology has steadily advanced, anchored by the core standards of HTTP, URI, HTML, JavaScript and Cascading Style Sheets. Multiple media types have converged under the Web umbrella, including video, audio, vector graphics and animation, ably supported by JavaScript and CSS. Maps have also become commonplace on the Web, although they have not been incorporated into HTML directly, while other media types such as video, audio and vector graphics have recently been added.

1.1 How Web Standards Evolve Today

The early Web was characterized by intense techno-political drama during which the fortunes of companies rose and fell based on their "control" of Web standards such as HTML and others.

The social revolution of "Web 2.0" has changed the dynamics of standards development. Individuals and groups have the opportunity to collaborate closely and more productively, using development tools which allow the best technical and social solutions to surface and thrive.

The W3C provides a type of public forum it calls a "Community Group" (CG), which comes in the form of Web resources devoted to supporting the 'socialization of ideas' in an intellectual property-appropriate environment, prior to standardization. The "Maps For HTML Community Group" (Maps4HTML), was formed as a result of bar camp discussions at the joint W3C - OGC 2014 workshop "Linking Spatial Data". The objective of this CG is to specify and prototype one or more map media types, together with a Web Components-based client. Participation in the Maps4HTML community is open and free. It is a requirement of contributors to agree to the Contributor License Agreement [19], which grants certain contribution copyright and patent license considerations to the community, in the interest of creating unencumbered standards. This requirement has possibly hindered the flow of contributions to the specifications; however the membership in the group is slowly growing, as is the list of influential Web map developers participating in the Maps for HTML organization on Github.

The Maps4HTML CG is trying to emulate the successful approach pioneered by the Responsive Images Community Group (RICG), in influencing the standards process. The RICG efforts have recently resulted in a new elements and attributes in the HTML standard being implemented by browsers, which support serving different images to devices of differing capability and screen size.

The evolving standards known collectively as "Web Components" provide the opportunity today to do more than simply propose new paper standards: they afford us the ability to implement proto-standards such that the Web at large has the chance to 'kick the tires before buying'.

The goal of the present work is to provide the community with a working implementation of a proposed standard media type and a (Web Components) Custom Element, which together could pave the way to extend Web standards to include modern Web maps, thereby improving the Web platform.

2. Motivation, Use Cases and Requirements

2.1 Motivation

The motivation for Maps for HTML is socio-economic. Free markets are essential to optimal resource allocation [3] and adaptation to change. Fair and decentralized access to the information required to participate in markets is the key to their effective functioning [5]. It has been shown for example that better communication technology can improve welfare for participants in a free market [8]. Location is widely acknowledged to be important or central information in many if not most market transactions [3][4][5]. It is the idea of the broad and deep importance of location information which has driven the development of Spatial Data Infrastructures by governments across the globe, which in turn helped engender the concepts and principles of Open Data. The Web has emerged as the principal standard information communication technology of modern civilization; therefore for the social and economic benefit of all, location must be a standard, decentralized component of the Web's fabric, and this remains (still today) an open problem to be solved:

> *"The fact that the method by which such knowledge (viz. the circumstances of time and place) can be made as widely available as possible is precisely the problem to which we have to find an answer". [5]*

The barriers to creating and using maps on the Web are too high, and prevent knowledge from being as "widely available as possible". In the first place, a barrier exists due to exclusive reliance on JavaScript, contravening the established accepted principles and good practices of the Web [6][9][2]. Furthermore, each of the many JavaScript mapping libraries has its own characteristics, dependencies and learning curve, such that to "make a map" has no common starting place or technique, and is at its core a non-standard experience that is and will remain un-crawlable. Map projections, which are a core part of effective communication with maps, are essentially out of reach for all but those with expertise in advanced or proprietary map scripting techniques. Map feature symbolization today is essentially a server-side activity, and remains mostly outside the domain of seemingly relevant Web standards, for example CSS and SVG.

Evidence of the 'centralization' of maps on the Web can be observed in the results of mainstream Web searches: whereas there may be billions or millions of hits for a given search under the 'All', 'Images' or 'Videos' categories, there is at most a single map found under the 'Maps' category, regardless of search provider. The map is also usually sourced from the search provider.

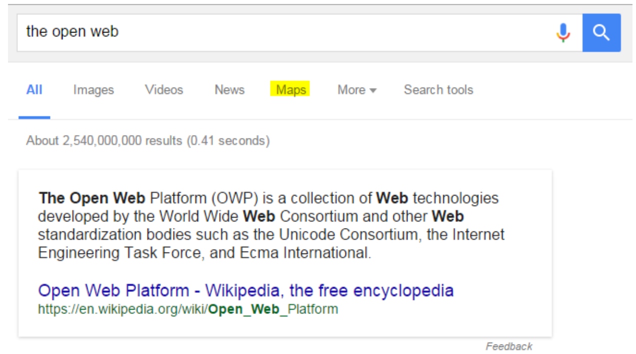

Figure 1 Web search results include one map at most

Over time, evolving Web standards have progressively lowered the barriers to authoring and using various forms of media: text, images, drawings (vector graphics), video, and audio have all been gradually incorporated into core Web standards. Maps should be the next media type to find its place with them.

For maps on the Web, the challenge in actually lowering barriers lies in developing a standard that really is simpler than what already exists, is applicable and leverages existing successful standards, *if possible* [10]. Based on the principle of information re-use [9], it is logical to conclude that the objective of a new or rejuvenated HTML <MAP> element supported by other Web standards, such as URI, MIME etc. would represent an lower barrier than existing Web map techniques, especially when considerations of search crawlers are included. What remains is to document use cases and derive requirements for the element.

2.2 Use Cases and Requirements

Based on arguments described under Motivation (2.1), the Maps For HTML Community Group began from the premise that extending HTML to enable basic map functions via declarative markup was desirable. It became clear, based on the experience of other Community Groups, notably the very popular Responsive Images Community Group (RICG), that development of "solutions" to problems should proceed based on credible use cases and their resulting requirements. As a result, the Maps for HTML community group developed a Use Cases and Requirements [12] document to be as generic as possible without pre-identifying a preferred "solution".

One of the core use cases derives directly from the motivation, namely extend HTML to allow an author to create a standard Web map declaratively, while simultaneously supporting fallback processing geared to older or non-compliant browsers. To accomplish this, it is thought that the behavior of the existing <MAP> element could provide the fallback processing, while the new behavior could be implemented by conforming user agents. This would allow Web crawlers to create page indexes which could categorize them as being in the 'Maps' search result category noted above, while ranking them based on location as well as other criteria common to search algorithms.

2.3 Development methodology

A prototype is more valuable than a slide presentation, when it comes to standards proposals. To succeed in today's Web standards environment, a specification must evolve in concert with one or more implementations. Web Components is itself a set of emerging standards allowing the development of custom elements, that is, elements which implement behavior that is established by JavaScript, HTML templates and CSS. The approach adopted by the Maps For HTML Community Group has from the beginning emphasized the importance of prototyping.

Instead of using a "waterfall" method of development of first writing a complete specification and then a developing a full implementation of that document, the project has adopted an iterative approach. In this approach, a single feature is added to the specification(s) and then developed in the implementation. By keeping the specification more or less in sync with the implementation, it is hoped that a more coherent experience for readers and users will be achieved.

2.4 Results: The <web-map> Custom Element

Web mapping is commonplace today. Web maps are performant and comprehensive, and Web users are sophisticated in their application of Web maps and demanding in regard to performance and features. A declarative implementation of Web maps thus has a very high standard to meet. The application for this project was developed as a Custom Element, using the excellent and popular Leaflet.js and Polymer libraries. The element is available for use or downloaded via Github [13], and is shown below (Figure 2,Figure 3,Figure 4).

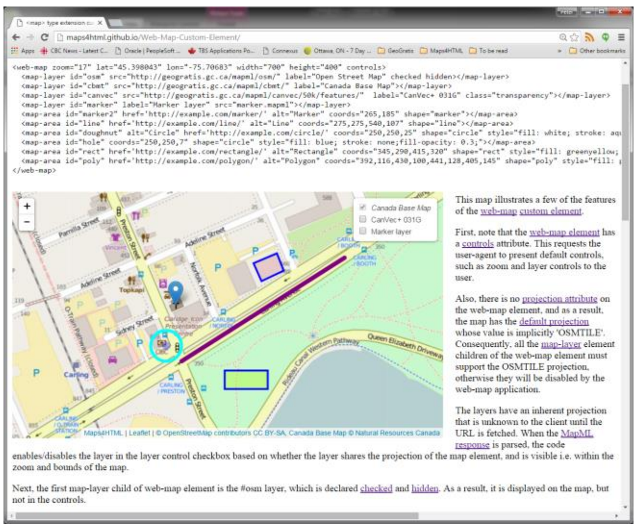

Figure 2 The <web-map> Custom Element

Figure 2 depicts a typical Web map displayed in the default projection of the Open Street Map tile system. Other features of that map include multiple image and vector layers, default controls, and CSS styling of HTML author-created hyperlinks.

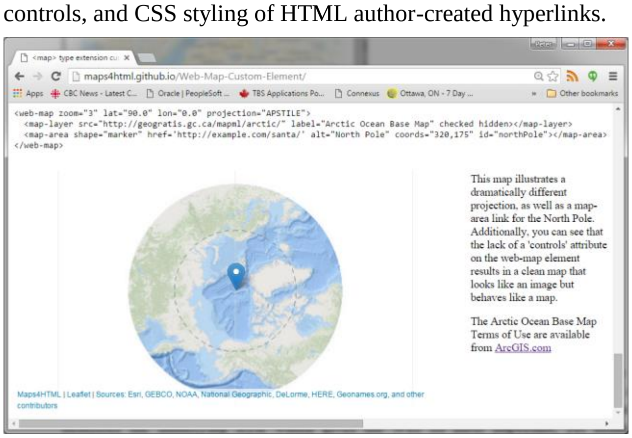

Figure 3 The <web-map> Custom Element supports diverse projection systems – Alaskan Polar Stereographic shown

Figure 3and Figure 4 show Web maps in non-default projections, including Polar Stereographic and Lambert Conformal Conic, achieved through simple projection attribute declarations.

Section 3 below discusses the techniques required to achieve this simple HTML author interface.

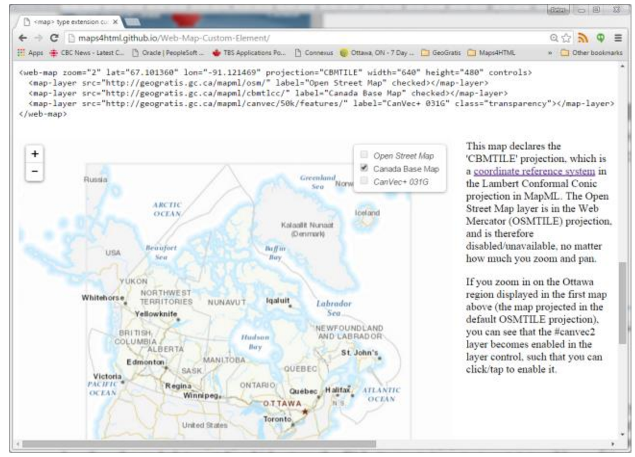

Figure 4 Layer projections must be compatible. Lambert Conformal Conic shown.

3. Web map components and design

3.1 MapML - Map Markup Language

Among the requirements to support maps in HTML, there are several which drive the design of the MapML specification [14], and indeed give rise to the requirement for the format itself. The first and perhaps most important element of the design of MapML was the application of some of the media type design considerations described by Representational State Transfer (REST) [2]. REST was considered important in the design because it describes the underlying style which gives the Web it's most important, yet subtle qualities, such as URL/domain - independence. One such quality is described by Tim Berners-Lee in his book, "Weaving the Web" [16]: "This important fact (the domain-independence of URLs) enabled a huge diversity of types of information systems to exist on the Web. And it allowed the Web to immediately pick up all the NNTP and FTP content from the Internet". Similar considerations apply today to MapML, in that there are vast quantities of spatial data sources made available as tile caches, and via WMS image servers, which can be used as sources by MapML servers in a similar way as the early Web 'rode' existing FTP and NNTP content to mainstream adoption and success. MapML includes markup constructs (forms and links) which afford interactions between client and server in such a way as to allow the client to request map documents which fulfill the map display needs of the client, but in contrast to tile caches and WMS services, do not require specific URL patterns or parameters. Additionally, MapML documents have a defined scale, extent and projection; these properties are considered part of the fundamental semantics of maps, and are therefore part of the definition of the media type. Clients and servers agree, via the media type definition, on self-describing message semantics [11] and as a result, representative maps on the client are enabled. These semantics (including among others: projection, scale, and extent) are as important to correct map interpretation as character encoding knowledge is to correct text interpretation and display.

3.2 Tiled Coordinate Reference Systems

MapML is a hybrid raster/vector hypermedia format. One of the core characteristics of modern Web map services is that they serve map tile images in standard formats. The tiled images are composited on the client to achieve a continuous map-like experience at successive discrete scales ("zoom levels") of rendering. In today's Web maps, the understanding of the tile cache URL space is implicitly shared by clients and servers, in that the tile cell ‘coordinate system’ is used to define the URLs of the tile cache, while the tile images’ pixel origin and resolution in a specific map projection’s units of measure, as well its axes’ orientations and transformations are implicitly ‘understood’ by the client. If such assumptions were not possible, Web mapping as we have come to know it would not be possible. MapML defines the coordinate systems, zoom levels, pixel origin, resolution, axis orientations and tile sizes as part of the media type definition, under the rubric of “Tiled Coordinate Reference Systems”. In this way, clients need not understand anything about the semantic structure of the URL space of the server; they must only rely on and agree to the definition of the named/identified coordinate systems as found in the media type definition. This shared understanding through a media type is possibly the key factor that has yielded the astounding success of the Web (the HTML media type definition, specifically), and re-using this architectural style is considered essential to the potential for future success of the geo-Web.

```
▼<mapml>
  ▼<head>
    ▼<title>
        Tile references for extent@value, sorted by distance from centre
      </title>
      <meta http-equiv="Content-Type" content="application/xml"/>
      <meta charset="utf-8"/>
      <meta name="projection" content="OSMTILE"/>
      <meta name="zoom" content="15"/>
      <meta name="area" content="20"/>
      <base href="http://geogratis.gc.ca/mapml/osm/"/>
      <link rel="license" href="http://www.openstreetmap.org/copyright" title="©
      OpenStreetMap contributors CC BY-SA"/>
    </head>
  ▼<body>
    ▼<extent units="OSMTILE" action="http://geogratis.gc.ca/mapml/osm/" method="get"
      enctype="application/x-www-form-urlencoded">
        <input name="xmin" type="xmin" value="2429536.0" min="0.0" max="8388608.0"/>
        <input name="ymin" type="ymin" value="3004363.0" min="0.0" max="8388608.0"/>
        <input name="xmax" type="xmax" value="2430674.0" min="0.0" max="8388608.0"/>
        <input name="ymax" type="ymax" value="3005003.0" min="0.0" max="8388608.0"/>
        <input name="zoom" type="zoom" value="15" min="0" max="18"/>
        <input name="projection" type="projection" value="OSMTILE"/>
      </extent>
      <tile col="9492" row="11737" src="http://b.tile.osm.org/15/9492/11737.png"/>
      <tile col="9492" row="11736" src="http://c.tile.osm.org/15/9492/11736.png"/>
      <tile col="9493" row="11737" src="http://a.tile.osm.org/15/9493/11737.png"/>
      <tile col="9493" row="11736" src="http://b.tile.osm.org/15/9493/11736.png"/>
      <tile col="9491" row="11737" src="http://c.tile.osm.org/15/9491/11737.png"/>
      <tile col="9491" row="11736" src="http://a.tile.osm.org/15/9491/11736.png"/>
      <tile col="9492" row="11738" src="http://b.tile.osm.org/15/9492/11738.png"/>
      <tile col="9492" row="11735" src="http://c.tile.osm.org/15/9492/11735.png"/>
      <tile col="9493" row="11738" src="http://a.tile.osm.org/15/9493/11738.png"/>
      <tile col="9493" row="11735" src="http://b.tile.osm.org/15/9493/11735.png"/>
      <tile col="9491" row="11738" src="http://c.tile.osm.org/15/9491/11738.png"/>
      <tile col="9491" row="11735" src="http://a.tile.osm.org/15/9491/11735.png"/>
      <tile col="9494" row="11737" src="http://b.tile.osm.org/15/9494/11737.png"/>
```

Figure 5 MapML document showing tile encoding

Simplicity is an important factor in the success of any technology, and it is an Internet axiom [17] that any successful complex systems have evolved from a successful simple system. To that end, MapML is an HTML-like format based on MicroXML. MicroXML [18] was designed by the MicroXML Community Group to be an HTML-compatible, simplified variant of UTF-8 - only XML; one which eliminates most complexities of the XML specifications, notably namespaces.

As mentioned above, MapML also supports vector spatial information. The feature and geometry definition of MapML vectors is adapted from the GeoJSON format, which in turn adapts the feature and geometry model of the Simple Features Specification for SQL [7]. MapML is intended to be compatible with standard Web browser implementations. Its MicroXML syntax allows it to be processed as a document using DOM + CSS, while the feature geometry model allows for the definition of WebIDL-based client APIs using the JSON syntax familiar to HTML authors. The vector data model of MapML is similar to the raster model in that MapML documents which contain vectors also have a defined extent, zoom level and projection. It is up to the server to serve data that is of the appropriate resolution for a given zoom level.

Finally, the incredible scale of the modern Web is often attributed to its support of simple links. To create a Web of maps, it is very desirable to copy the characteristics of the Web that have enabled its successful growth. Thus, not only should it be possible to declaratively create a map in a Web page, but that map should be able to link to other maps, and so on, leveraging the federation power of the Web as a (potential) whole in describing spatial ‘reality’. Furthermore, no prescription should be made, outside of that of the “uniform interface” described by REST, which concerns the form or semantics of URLs used by maps to link to other maps. In this way, a map might link to images or other spatial formats as overlays, images or HTML as legends, or in fact other MapML services / documents as larger-scale or adjacent same-scale maps to be presented on demand to the user. Although MapML has schema constructs to support such links, it remains an area of mostly unresolved promise for future development. Implementations are needed to provide link targets for the creation of a nascent geo-Web.

3.3 The MapMLServer

There are two separate components in this server side Java “Web application” currently, in which functionality is allocated between the components based on the raster or vector nature of the content being served. The first component is the MapMLTileServlet, which is a Java servlet that can be configured to act as a MapML-serving proxy for a standard tile ‘cache’. The servlet is configured via web.xml init-param servlet variables, (including a URL template for tile resources), to serve MapML documents containing <tile> elements with the actual URLs for the tiles which are needed to respond to a Web request over a given map extent. This servlet effectively replicates on the server some of the logic currently found only in client libraries (e.g. Leaflet or OpenLayers). The processing that the servlet performs to respond to a request is minimal, and it is relatively fast. It is the tile cache server that carries the bulk of the workload. A typical response is shown below.

```
▼<mapml>
  ▼<head>
    ▼<title>
        Natural Resources Canada's CanVec+ 031G - Map Markup Language
      </title>
      <meta http-equiv="Content-Type" content="application/xml"/>
      <base href="http://geogratis.gc.ca/mapml/canvec/50k/features/"/>
      <link rel="stylesheet" href="../../../styles/canvec_cantopo_svg.css"
      type="text/css"/>
      <link rel="license" href="http://www.nrcan.gc.ca/earth-
      sciences/geography/topographic-information/free-data-geogratis/licence/17285"
      title="CanVec+ © Natural Resources Canada"/>
    </head>
  ▼<body>
      <!-- projection=WGS84 -->
      <!-- zoom=18 -->
    ▼<extent units="WGS84" action="http://geogratis.gc.ca/mapml/canvec/50k/features/?
      entry-type=full" method="get" enctype="application/x-www-form-urlencoded">
        <input name="xmin" type="xmin" value="-75.70842862129213" min="-76" max="-74"/>
        <input name="ymin" type="ymin" value="45.3953197238206" min="45" max="46"/>
        <input name="xmax" type="xmax" value="-75.70053219795227" min="-76" max="-74"/>
        <input name="ymax" type="ymax" value="45.39912045308345 5" min="45" max="46"/>
        <input type="zoom" value="18" min="15" max="18" step="1"/>
        <input type="projection" value="WGS84"/>
      </extent>
    ▼<feature id="6b5d3030fa2640c7ade3e8cc368ed72d" class="_1481312">
      ▼<properties>
          <accuracy>5</accuracy>
          <code>1481312</code>
          <definition>4</definition>
          <id>6b5d3030fa2640c7ade3e8cc368ed72d</id>
          <isolated>0</isolated>
          <lakgeodb>ONTARIO GEOGRAPHIC NAMED EXTENT LAYER</lakgeodb>
          <laklang>-1</laklang>
          <laknameen>Dows Lake</laknameen>
          <laknamefr>Dows Lake</laknamefr>
          <laknameid>3c139b39c6ce11d892e2080020a0f4c9</laknameid>
          <permanency>1</permanency>
          <rivlang>0.0</rivlang>
          <theme>HD</theme>
          <valdate>198805</valdate>
        </properties>
      ▼<geometry>
        ▼<Polygon>
          ▼<coordinates>
              -75.7004373 45.3910771 -75.7009141 45.391092 -75.7013957 45.3911071
              -75.7013962 45.3911431 -75.7013484 45.3914135 -75.7013647 45.3916924
              -75.7017012 45.3920504 -75.7021009 45.3923449 -75.7022552 45.3924249
```

Figure 6 MapML document showing feature encoding

The other component of the MapMLServer, which serves vector information, is also a servlet. This component uses open source spatial database technology to serve vector features encoded in MapML. The nature of such processing is inherently resource-intensive, and it is therefore not as fast as simply serving tile references. A limited subset of the "CanVec+" open data product content serves as a proof-of-concept for vector processing in the MapML format.

3.4 The HTML <web-map> element

Although it represents a good simple starting point for future progressive enhancement, the historical <MAP> element is evidently insufficient in consideration of modern Web mapping requirements. A custom HTML element named "web-map" [13] was co-developed with a specification [15] for its syntax and semantics. The syntax and function of the <web-map> element is in fact an extension of core aspects of the existing <MAP> element (see in particular Figure 2). When the actual HTML standard is subject to revision on this matter, it should be a matter of discussion among browser developers and the Web communities to resolve whether to support progressive enhancement of the existing <MAP> element to the include the new semantics, or to create a completely new element devoted to Web mapping.

The <web-map> element specification inherits features from different sources. The first influence is from the Leaflet.js map library's L.Map object type. The lat, lon, and zoom attributes reflect analogous L.Map options for locating the initial center of the map. The optional projection attribute, for which an unspecified value is considered equal to "OSMTILE", is necessary to enable non-default map projections required by standard cartographic practice. The Boolean "controls" attribute is used to tell the user agent to present default-styled map controls, such as pan/zoom, layer controls, etc. Finally, the width and height attributes are similar to attributes of the same name for the HTML <img> element, allowing the HTML author declarative control over the dimensions of the map.

The <map-layer> child element of <web-map> is used to identify a source of map information for the map. In this way, it is similar to the <source> element used by HTML's <video> and <audio> elements. The required src attribute is an opaque URL to a MapML document or service. In contrast to the <video> or <audio> elements' child <source> elements, the <map-layer> sources of map information are not mutually exclusive; successive <map-layer> child elements are rendered on the map if possible, in painter's order. The <map-layer> may carry the optional Boolean attributes "checked" and "hidden". "checked" means that the layer should be drawn on the map if appropriate. If no "checked" attribute exists, the layer will be added unchecked to any layer control that may be present. To be rendered, the layer must be discovered to be within the scale and extent bounds of the map, and to share its projection (Figure 4). Layers can thus be added to the map and be turned on or off by the user via controls or by the HTML author via script behavior.

The <area> element child allowed as <map> element children designate links of different shape types. This design facet is reflected in the <map-area> element child of the <web-map>. The <map-area> element represents features of different types drawn on the map, at the map's declared zoom level, which can be designated as link anchors. The pixel-based coordinate system used by the HTML author to locate these features is established via CSS and/or the width and height attributes of the parent web-map element. "Pushpin" maps are a common use case for Web maps, and this use case is supported by <area> element's extended shape value of "marker" (Figure 2).

3.5 <web-map> API

Web mapping is characterized by a diversity of requirements specific to the intended use or application of the map being designed and built. Like many elements in the HTML family, the web-map element needs a scriptable API to address such extended requirements. Scripting allows maps to be progressively enhanced by the user agent in a way which delivers an enhanced map user experience to users on advanced or capable devices, while in principle permitting the core experience to all users on as many devices as possible.

Adoption of the web-map prototype element and its use in real-world mapping applications will certainly help identify specific needs for development and further elaboration of an API. In the meantime, a simple set of behaviors are specified (in WebIDL) and implemented by the custom element as a first-iteration API.

First, a named constructor [WebMap(width, height, lat, lon, zoom, projection, controls)] is provided for the web-map element, which allows a map to be created and populated on-the-fly by script, and inserted into the DOM at a document location chosen by the script. <map-layer> and <map-area> child elements can be created using the standard DOM createElement method, and their attributes can be established using standard DOM methods such as setAttribute. These elements can be appended, inserted and removed in the <web-map> element content, and doing so allows map layers and links to be created and removed dynamically. Additionally, controls can be added to or removed from the map at run time, by toggling the controls attribute of the web-map element with script. The size of the map can be dynamically changed by manipulating the width and height attributes. Currently, there are no means whereby the projection of the map can be changed after the map's creation. The reason for this is that tile-based map layers do not lend themselves to on-the-fly projection, and it is the map-layer children of the web-map element which are required to conform to the parent element's projection attribute value. If dynamic map projection is a requirement, it is thought that dynamic map creation in a different projection, replacing the original map in its location, scale, size, and document position will be the preferred technique. Experimentation is required in this area to establish a strong pattern to respond to the need.

The next important characteristic of the web-map API is that the lat, lon and zoom attributes of the element are updated in the DOM as the user controls the map, reflecting their dynamic values, i.e. the values due to the interaction with the map, not simply the initial values given them by the HTML author. In this way, scripts can read and use the location and scale of the map for the purposes of enhancing the user experience – possibly by displaying those values in a cut-and-paste text box, for example. While the lat, lon and zoom attributes are read-only to scripts, the web-map element API provides the zoomTo method, which allows the script author to change the location and / or the scale of the map based on other information – including for example, the location of the device based on the results returned by GeoLocation API calls.

Finally, the web-map element exposes read-only HTMLCollections of its layers and areas via the .layers and .areas properties. These are convenience properties that could also be accessed using standard DOM query and navigation methods.

4. Conclusion

In conclusion,

a) Decentralized Web map information is essential to society
b) its architecture is custom-built for decentralization of information, and is based on standards

c) Web maps do not follow Web architecture and standards, and in consequence have become centralized and/or the domain of specialists

Please join the Maps for HTML CG to openly and collaboratively develop new Web standards and implementing software which:

a) Reflect the importance of decentralized maps and locational knowledge by extending HTML to include maps
b) Propose a new document standard conforming to Web architecture, leveraging some existing patterns (tiles, WMS, Simple Features) but which favours decentralization (hypermedia, uniform interface, self-description)
c) Progressively surfaces capabilities for Web maps, starting with a) above, but extending to sophisticated GIS functionality relying on proposed browser APIs and the code-on-demand constraint of Web architecture

5. ACKNOWLEDGMENTS

The author would like to thank Benoît Chagnon and Norah Brown for their patience in discussing many of the ideas in this document. The author thanks as well the anonymous reviewers from the Location Web 2016 workshop who provided valuable feedback. All errors and omissions obviously remain the responsibility of the author.

rmation services using World-
eedings: Selected papers of the
de Web, Journal: Computer
Volume 27 Issue 2, Nov. 1994
.doi.org/10.1016/0169-

tectural Styles and the Design
rchitectures. Doctoral
lifornia, Irvine, 2000.

etrieved 2015-02-08. Low
Free_market#Low_barriers_to_

w of Geography – retrieved
Tobler%27s_first_law_of_geog

f Knowledge in Society. The
Volume 35, Issue 4 pp 519-

ancement – retrieved 2016-02-
iki/Progressive_enhancement

[7] Open Geospatial Consortium, 2006. OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 2: SQL option. 06-104r4 http://www.opengeospatial.org/standards/sfs.

[8] Jensen, R., 2007. The Digital Provide: Information (Technology), Market Performance, and Welfare in the South Indian Fisheries Sector. The Oxford Quarterly Journal of Economics, Volume 122, Issue 3. Pp 879-924. http://qje.oxfordjournals.org/content/122/3/879.abstract

[9] W3C Consortium Technical Architecture Group finding, 2006. The Rule of Least Power. http://www.w3.org/2001/tag/doc/leastPower.html

[10] W3C Consortium HTML WG Working Draft, 2007. HTML Design Principles – *Pave the cowpaths; Do not Reinvent the Wheel*. http://www.w3.org/TR/html-design-principles/#pave-the-cowpaths

[11] W3C Technical Architecture Group finding, 2013. Authoritative Metadata. http://www.w3.org/2001/tag/doc/mime-respect.html#abstract

[12] The Maps for HTML Community Group, 2015. Use Cases and Requirements for Standardizing Web Maps – Draft Report. http://maps4html.github.io/HTML-Map-Element-UseCases-Requirements/

[13] The Maps for HTML Community Group, 2015. The Web-Map Custom Element – Open Source Jaf o r(les)4(th)-8rinjec13((h)-6(EMC
/Span <</Lang (en-US)/MCID 64 >>73C 0 0 1 rg0 -

(1)-6(3)-6(5)]TJ/TT1 9 Tf15.38 0 Td( )Tj/TT0 9 Tf2.64 0 Td[(T)11(h)-6(e)4( )-2(M)-3(a)4(p)-6(s f)12(o)-6(r)-3(HT)13(M)-3(L)11( )-2(Co)-5(m)18(m)18(u)-6(n)-6(it)-