

# Free and Open Source Software for Geospatial (FOSS4G) Conference Proceedings

---

Volume 16 *Bonn, Germany*

Article 8

---


2016

## Semantic Markup for Geographic Web Maps in HTML

Malte Reißig

*Leibniz Institute for Regional Geography (IfL), Leipzig, Germany*

Follow this and additional works at: <https://scholarworks.umass.edu/foss4g>

 Part of the [Geographic Information Sciences Commons](#), and the [Other Computer Sciences Commons](#)

---

### Recommended Citation

Reißig, Malte (2016) "Semantic Markup for Geographic Web Maps in HTML," *Free and Open Source Software for Geospatial (FOSS4G) Conference Proceedings*: Vol. 16 , Article 8.

DOI: <https://doi.org/10.7275/R52V2D9J>

Available at: <https://scholarworks.umass.edu/foss4g/vol16/iss1/8>

This Paper is brought to you for free and open access by ScholarWorks@UMass Amherst. It has been accepted for inclusion in Free and Open Source Software for Geospatial (FOSS4G) Conference Proceedings by an authorized editor of ScholarWorks@UMass Amherst. For more information, please contact [scholarworks@library.umass.edu](mailto:scholarworks@library.umass.edu).

# Semantic Markup for Geographic Web Maps in HTML

Malte Reißig

Leibniz Institute for Regional Geography (IfL), Leipzig, Germany – m\_reissig@ifl-leipzig.de

**KEYWORDS:** Geographical Web Maps; HTML; Schema.org; Dublin Core;

## **ABSTRACT:**

*In the recent years more and more geographical web maps have been developed and published on the Open Web Platform. Technically this has turned all variants of these maps into documents of the Hypertext Markup Language (HTML) making them appear to us naturally as graph-like and semi-structured data. In this dispute with geographical web maps and HTML we draw on the notion of so called “map mashups”. Requiring an alternative model and definition of what such a map is, our research allows us to build and refine supportive technology which helps us in analyzing and interpreting information map makers code into their visualizations. The spectacles we take on to shine light on the current authoring practices behind many geographical web maps are informed by the perspective of a “critical map reader”. A task-oriented conception of “map critique” helped us to deduce a meaningful user perspective from which we specifically call the semantic web community for support on how to represent various information presented in maps from many authors and sources. With this perspective and questions in mind we investigated the Schema.org vocabulary as an ontology to use for turning elements of geographic web maps into textual statements referencing entities in the “outer world”. To illustrate and to make our investigation of the corresponding web standard documents easily applicable for map makers, to open up the discussion, but also to challenge and develop our first conclusions, we implemented them as a minimal extension to the standard API of the LeafletJS open source web mapping library.*

## **1 Motivation**

As investigations into the “evolving web mapping technologies” by Roth et al (2014) highlight, more and more geovisualizations are developed and published as HTML Documents. For this report we investigated arbitrary LeafletJS based examples simulating “map mashups” (see Turner, A. J. 2006, Gartner, G. 2009) to think about the semantics of the markup generated, especially those semantics concerned with terminology from the geographic domain. Through Bittner et al we receive our focus on advancing the semantics of map mashups, as the number of geographic web maps published on the world wide web seems to difficult to keep track with. Map mashups on the web, they describe, are essentially “users mixing information with so called base maps through geo-referencing” (after Roth/Ross 2009 and Crampton 2010, see Bittner, C.; Michel, B. 2013, p.112). This report is inspired by the publications of Roth, R. E. (2013) and Schiewe et al which illustrate how a more “wholistic understanding of map usage” (Schiewe, J.; Schweer, M. 201, p. 9) can inform cartographic research. Following a user-centered investigation of semantic markup for geographical web maps, we argue that it is possible to make “critical map reading” an explicit, interactive and possibly even engaging part of the everyday geographic web map experience. Up to my knowledge there is no research reported yet which investigates and refines geographic web maps as what they also have become: documents of HTML. I do so in the hope to equip future map makers and publishers with some knowledge about semantic markup and therewith contribute to a more responsible publishing of geographic web maps, a publishing which not oly considers the map readers perspective but accounts for that through providing advanced accessibility and transparency on information composed in its maps.

## **2 Introduction to HTML and geographic web maps**

When we surf geographical web maps on the world wide web we find out that we can understand them as a thing of composite structure, authored, represented and described in documents of the Hypertext Markup Language (HTML). Strikingly, we could not find any description of what a meaningful markup of a geographic web map could be viewed as, thus this investigation. If we understand geographic web maps as composites it

becomes clearer that a HTML document enclosing such a map aggregates many documents, not just one. Through utilizing web mapping libraries map makers effectively turn geodata of various formats into HTML elements while that is a "domain specific" vocabulary and "mixed markup language". This acknowledges that its a "formal language" used to encode rules for presenting and describing data (see, for example Wikipedia Community 2016). But how does this language reflect our current practices and what are current possibilities to adapt the HTML vocabulary?

Advancing our markup through bringing in elements and attributes with meaning anchored in the geographical domain would allow us to make geographical conceptions and data contained in each map explicit and accessible. Having these values accessible would us to build supportive tools and visualizations to match our current web mapping practices.

To define the scope of this research and to define what I would call a valuable and improved document structure for a geographical web map is, it must help to give answers about WHO states (or stated) WHEN WHAT about WHOM or WHAT in this map. So, for analyzing maps we rely on questions like: Who contributed to this map? What topics are represented in this map? What areas of the world does the map explicitly deal with? When was this map made? We can easily deduce these questions from course materials on "critiquing maps" by Mattern, S. C. (2016) as well as from the MediaSmarts (2016) guide to "Deconstruct Webpages" for the "7th-10th Grade". With that, it seems logical to state that what is missing to answer these questions are data values accessible in HTML concerning each basic element (Popup/Detail, Marker/Feature or Layer/Geometry) in a map mashup, covering at least an elements Title, Caption, Source, Attribution and Date. In some case these values may already be available in the archive, library or public database providing map makers their information or geodata but yet there does not seem to exist an interface these values could be passed on to.

## 2.1 Typing of relevant geographic web map elements with Schema.org

To express more specific semantics than those defined by the HTML Standard the W3C and others developed HTML Markup Extensions which allow users of HTML to extend the HTML vocabulary into their domain without invalidating the HTML for interpreters. One W3C (2015b) recommendation is Schema.org. According to companies like Google that is well supported by "many major search engines" (2016). The available notations for the latter are Microdata (W3C, 2013), RDFa (W3C, 2015a) and JSON-LD (Sporny, M, Longley, D., Kellogg, G., Lanthaler, M., Lindstrm, N., 2010). To allow basic annotation like I would envision it, the vocabulary must contain terms and definitions for formalizing spatial data values inherent to each geographic web map. Furthermore, essential elements of a geographic web map should be annotatable as (a) distinct elements of information and (b) information representing a certain type of information (like City or Organization) and, one level higher, possibly even a concrete token or instance of such (like Leipzig or Leibniz-Institute for Regional Geography). If these needs could be met I would argue that semantic authoring of geographic web maps has been significantly advanced.

The World Geodetic Systems has become without a doubt a useful index to all kinds of information on the web but many map makers might not be aware of the fact that they are not directly exposing their information to it when creating a web map. To express values related to a specific geographic reference system we found that the geo (WGS 84) looks to be integrated with many other linked data vocabularies (<http://lov.okfn.org/dataset/lov/vocabs/geo>) but for us of too limited scope. The Schema.org vocabulary allows us to express Geo Coordinates (including elevation) while also allowing for more complex data definitions such as Geo Shape. As this investigation will show, when annotating basic elements of a geographic web map the elements mapped are connected to the WGS 84 reference system but this connection is not accessible when inspecting the HTML of, for example, the LeafletJS example (<http://leafletjs.com/examples.html>) documents. I therefore took a closer look at the geographic terms provided by the public Schema.org vocabulary:

- A Place is the basic geographic entity allowing map authors to note "entities that have a somewhat physical extension". This can be done through attributing them with a geographic area or point location (geo).
- When noting such a Place authors are alternatively capable of expressing a geographic reference through either specifying an address value (like a Postal Code) or through specifying the GLN (globalLocationNumber)
- Furthermore authors can annotate two types of hierarchical relations between Places, one relation type is to express places in which the one being annotated is containedIn (looking up the hierarchy), the other one to describes places contained by the one being annotated (containsPlace)

- The types Administrative Area and further down its hierarchy, City, Country, State, LocalBusiness, Landform or Civic Structure are all a more specific embodiment of Place but do not add any new attributes to the vocabulary.

Let us now exercise some statements integrating these terms with expressions about more general entities the schema.org vocabulary offers:

- When talking about Persons on our map, we can state their "place of birth" (birthLocation) and "place of death" (deathLocation) or we can state two "contact locations", one for "work"(workLocation) and one for "home" (homeLocation), where all of these expect a thing of type Place.
- When noting a specific person we could further use the "Contact Point" (contactPoint) attribute to specify a concrete location and furthermore specifying a dedicated contactType, (being a simple text value) allowing us to specify new or reference existing terms from other taxonomies.
- When talking about an Organization on our maps, we can note one or many locations of it but also be more specific, for example we could specify a Place relating to it as foundingLocation, its areaServed, a point of sale or a so called contactPoint.
- When talking about Creative Works in our maps we can specify the "location depicted or described in the work", through using contentLocation or we can note down the locationCreated where the work was created.

Schema.org is an extensible and public general purpose vocabulary and to join work on advancing it everyone is welcome to join the Schema.org Community Group. As of its latest version (2.2) it refined some geographic terms and properties as they introduced Geo Circle, allowing to encode a central location (geoMidpoint) with a distance value (e.g. in meter or feet) as geoRadius. As illustrated by this short exercise we can build on Schema.org to reach a couple of our goals: (1) classify map elements as distinct items, e.g. representing a Person or Event and (2) make machine readable statements about these items including references to their spatial applicability. Furthermore we can annotate each of the mentioned items with basic attributes of any so called Thing in Schema.org: Of which, for example name, sameAs, alternateName and description all help programs and user to identify a specific Organization represented in a map.

But what we cannot easily annotate each of our map element with Schema.org is data about our information source, attributing it properly or for example the time reference of the information or the original publisher and creator of the information or when it was last modified. Now, if we think semantic annotation of geographic web maps from a "critical map readers perspective this data is essential because it helps readers to identify, attribute and contextualize the presented information. Studying the standard document (2015b) has shown that if we would author this in terms of Schema.org we would need to cover every annotated map element (Thing) in a reference of a Creative Work (representing a user having placed a marker). The properties of a Creative Work mirror essentially what we would need to properly attribute the information behind the items we place on our map. For example, the City of Leipzig represented by a simple marker would be annotated indirectly through stating that the "main entity depicted by this work" (mainEntity) is the City of Leipzig. This circuitous usage of the for us relevant terms in the vocabulary, of course, could be implemented in a way that it is completely transparent for map authors and at the same time perfectly valid semantic markup for programs and users.

After introducing this extra level of abstraction we would get all the properties we need for extensively attributing each map element with essential data. Authors could easily informaton about the identity of the Authors, Contributors and Publishers, data about the timely applicability (Created, Published, Modified) as well as data about the usage rights they claim (License) for the information making up their map element. But when adding semantic annotations to a geographic web map we are not confined to use just one vocabulary, as long as we re-use existing (and at best, well supported) vocabularies we can claim to have enhanced the interoperability of our geographic web map documents to at least some degree.

## 2.2 Authoring attributions on map elements with Dublin Core

Further research about existing and popular general purpose metadata definitions has brought to light the Dublin Core Metadata Element Set (2012). The "Dublin Core" defines essential terminology for describing all kinds of information resources and is subject of, according to the projects wiki, "countless implementations" ([http://wiki.dublincore.org/index.php/User\\_Guide](http://wiki.dublincore.org/index.php/User_Guide)). The fifteen terms in the "Legacy Namespace" ([http://wiki.dublincore.org/index.php/User\\_Guide/Publishing\\_Metadata](http://wiki.dublincore.org/index.php/User_Guide/Publishing_Metadata)) are slightly more expressive than our

case demands but are in essence those terms a reader would need when critically analyzing elements of a geographic web map. The complete Dublin Core Metadata Element Set is made of: *contributor, coverage, creator, date, description, format, identifier, language, publisher, relation, rights, source, subject, title, type*.

Having compared these terms to the before mentioned class in Schema.org (Creative Work) we came to the following conclusions: The essential metadata elements (DCMI 2012a) match, up to one element (relation), the Creative Work type definition on Schema.org. The latter is more extensive in comparison with the Dublin Core Metadata Element Set but misses a property to express the time reference of an information, while spatial attribution is in principle possible with Dublin Core, the more extensive (and explicit, in terms of “syntactic” and not only “formal semantic”) specification for expressing spatial attributes is present in Schema.org.

For our case, laying out fundamental attributes for interactive geographic web map analysis through its elements, the markup of the following Dublin Core terms are not absolutely necessary: Format (MIME-Type), Referenced/Citation, Language, Subject/Keyword. That is because we are not (yet) concerned with annotating web map elements thematically (integrating other or more taxonomies) and we are neither specifically concerned about annotating references, nor additional sources, nor a potential file format behind a map element.

As a first conclusion of comparing the two vocabularies we can safely state that, by their extensiveness and definitions, it would be possible to markup all our map elements using only the Dublin Core Metadata Element Set if we bring in two additional DCMI Metadata Terms (DCMI 2012b), namely created and modified. The Schema.org type or class name could then be expressed using the DCMI type property because it explicitly allows for the use of “controlled vocabularies of a third-party”. The coverage attribute would allow us to encode the spatial applicability of our map element (according to additional but common and “well defined” syntactical rules and exposing values in WGS 84). A practical dis-advantage of this, up to my knowledge, is that implementing it this way our markup annotations would not be officially supported by many search engines or other companies building data integration services based on Schema.org (e.g. <http://link.fish>). Expressing geographic indicators (such as an exact position or area) in the coverage field as defined by Dublin Core lacks explicitness to be considered as well defined as the spatial dimension one can express when building on Schema.org.

### 2.3 Concluding our investigation of Schema.org and Dublin Core

Following our analysis the main advantages of Schema.org are manifested by it being supported by various third-parties, such as Google (2016) as well as providing a publicly existing, easily accessible general purpose vocabulary to type and identify map elements as representations of “real world” entities. For not having to annotated every map element as an instance of a Creative Work first, or use the auxiliary properties named `additionalProperty` or `additionalType` to express all facets of a map element, our suggestion is to allow for classifying map elements as a basic Thing (Schema.org) and to integrate the rest of the needed attributes using terms of the Dublin Core Metadata Initiative. To group these attributes into a coherent statement about a map element we will use a specific HTML which integrates the expressions building on the two vocabularies.

To annotate geographic map elements with Schema.org we would use the properties: Type, Name, Description, sameAs, URL, as well as all forms and properties of type Place, GeoCoordinates and GeoShape. To markup our map elements in a way necessary by our use case we suggest to use the following terms and elements from the Dublin Core namespace: *Creator, Contributor, Publisher, Rights, Date, Created, Modified* and *Source*. Additionally we allow to express a Format and Language. Thus, using *Subject, Coverage, Referenced, Relation* and *Type* from Dublin Core is regarded as not essential.

## 3 Employing HTML Standard Elements for semantic annotation

For bundling our statements and relating them all to single items we must rely on standard HTML Elements to keep our document valid and machine readable. To do so we investigated the existing markup generated by the open web mapping library LeafLetJS and compared it to recommendations and definitions in the latest HTML Standard. When reading the HTML LeafletJS generate the `div` element is used as the only HTML Element to structure the map content. It is used as the HTML standard suggests, “with the class, lang, and title attributes to mark up semantics common to a group of consecutive elements”. But these `div` elements should be used “only when no other element is suitable” (W3C, 2014d).

To chunk our geographic web map into piece one could either consider using the *figure* element (though it should be used to markup “self-contained content” and not necessarily independent elements of content”) along

with its corresponding *figcaption* or, for example, the article element. As you will see in section 4, we chose the latter for map elements wherever because the definition of article (see Table 1) matches exactly this case.

For semantic annotation regarding of essential information on each web map element also the HTML standard provides ways to express authorship in HTML documents. At very first there might be a meta attribute value for the name author to find out about a documents author(s) (meta name="author"). Furthermore there was also introduced an rel="author" attribute in form of a so called "link type" (W3C, 2014g) allowing us to specify an author relation between a document about the author and the authored one in question. Another approach is to use the "class" attribute on any HTML Element to note that the content within this element is information about the "author". We could build upon these and repeat such author statements throughout our documents, one for each element to be annotated in our geographic web map but information on how well supported this is is scarce<sup>5</sup> and in this case we were directed to rather rely on Schema.org.

What follows (see Table 1) is a report of our investigation on how to structure distinct entities of content in HTML from which two options appeared to be most appealing when grouping statements about map elements in HTML. When marking up content within an article element the address element should be used for expressing "contact information on the content marked up in the nearest article element" (W3C 2014b). When marking up content within a more generic element, like div for example, the standard suggests us (without further specification) to include information about the author into the footer, an element "representing a footer for its nearest ancestor sectioning content", "typically contains information about its section such as who wrote it, links to related documents, copyright data, and the like" (W3C, 2014f).

Table 1: Comparison of HTML Element definitions for marking up distinct elements of content in a geographical web map.

HTML Element	Standard Definition
div	"has no special meaning at all", "used with the class, lang, and title attributes to mark up semantics common to a group of consecutive elements", "should be used only when no other element is suitable, as an element of last resort" (W3C 2014d)
figure	"represents content, that is self-contained and typically referenced as a single unit", "content may have a caption, typically the figcaption element." (W3C 2014e)
article	"represents any independent item section or content", "content in the article element should be independently distributable or reusable" (W3C 2014b)
section	"section is a thematic grouping of content, e.g. chapter", "is not a generic container element" W3C (2014i)
blockquote	"represents a section that is quoted from another source", "content inside must be quoted from another source, whose address, if it has one, may be cited in its cite attribute." (W3C 2014c)
q	"represents some phrasing content quoted from another source", "content inside must be quoted from another source, whose address, if it has one, may be cited in the cite attribute" (W3C 2014h)

Following this comparison of definitions we can say that aggregated elements in a geographical web map might be as well, if not better, wrapped in a *figure* or *article* element as both define to expect information on authorship as their "child elements". In the latter case, for example, an *address* element should be used to express author information relating to the next article, of which many are allowed to exist within a document. Instead of relying on HTML Elements and *class* names to express our desired semantics the markup extension we already investigated seems to be more explicit and (up to our knowledge) better supported. Following the newest recommendation for integrating Schema.org would be to build a JSON-LD representation for our geographic web map in a dedicated *script* element as a fragment of the HTML Document.

## 4 Semantic markup for LeafletJS based web maps

To illustrate the results of the investigation up to here I chose to extend and revise a tool that nowadays actually generates HTML for publishers of geographic web maps. I selected LeafletJS because it is an open source web mapping library and with its license explicitly encourages adaptation, study and distribution. What makes this library stand out is that it is especially concerned with providing compatibility and interactivity for web maps across various browsers, including “web browsers on mobile devices” (Agafonkin, V., 2016).

Mapbox for example, a commercial geoweb service and base map provider building on OpenStreetMap, builds its web mapping applications upon LeafletJS (MacWright, T. , 2013). This fact suggests that using LeafletJS is efficient when wanting to build professional web apps which enable users to mash up data of various file formats with geographically referenced imagery (Mapbox, 2016).

Before we can start with our implementation we are forced to identify the for us most relevant parts of a geographical web map. To do so we look at the API and relate its core concepts to the HTML generated by it. If we look at the HTML LeafletJS produces we can see that a web map assembles various graphics and texts which are not yet meaningfully annotated and do currently not expose the inherent geographical information of the map to HTML. It is therefore safe to assess that information already assembled by the map maker, along with our knowledge about the geographic reference system in use, is completely lost during the translation of the “logical map” (as constructed by users of the web application building on the LeafletJS API) and the resulting document. What is also obvious that some separation and structuring of the content is nonetheless manifested in the container elements generated by the LeafletJS developers.

Within every single map container of any LeafletJS (Version 0.7.3) based web map we found two basic elements called panes, namely a Tile Pane and an Objects Pane. While the former contains a so called Tile Layer which houses geo-referenced Tiles (Raster Graphics), the latter is created as a container for all Overlay, Marker- and Popup Layer. For our understanding of “map mashups” all elements on the so called Object Pane are essential. The former, the Tile Pane or often so called base map itself is out of scope for this work. A thorough investigation of a base maps structure and the therein encoded meanings and models needs to be a separate investigation with different terms.

If we additionally read out the wording of the LeafletJS API we find out about concepts of a Tile- or Base Layer, various other types of Layers but also about Markers and Popups. Furthermore Files play an important role as expressed through the integration of various file formats of the geoweb (e.g. Shapefile, GML, GeoJSON, TopoJSON) into the API.

The terms in which map makers who mashup and geo-reference information with imagery need to think of with LeafletJS could simple be understood and grouped Overlays. When creating overlays a user is concerned with geometrical elements like Points, Lines, Polylines or Polygons. Further we get to know that map makers place information regarding a very specific location using so called Markers, representing a pinpoint type of overlay. Using marker visually highlights and provides a layer of interactive to present additional (or on-demand) information to a so called “Place of Interest”. Additionally to image (file) based markers LeafletJS also allows for using vector drawn markers (namely through the class CircleMarker). The following sketch of the basic technical structure of a LeafletJS web map documents core items of the LeafletJS API. It is utilized here to illustrate the fact that even the simplest geovisualization produced with LeafletJS are of composite structure.

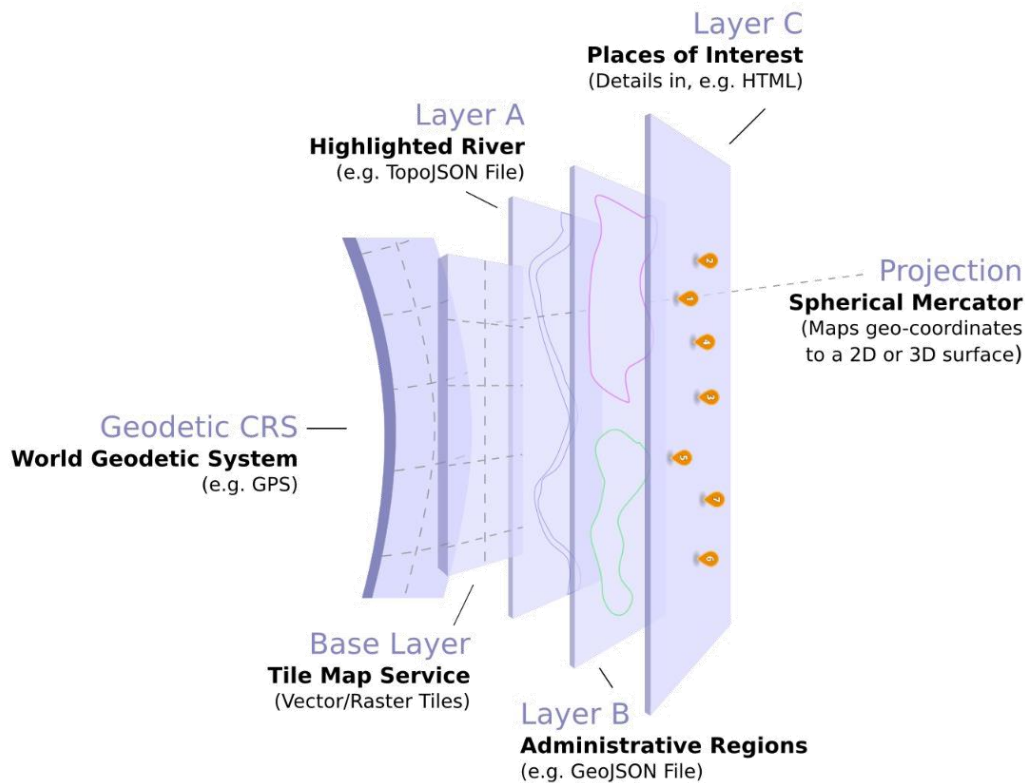


Figure 1. Core items of the LeafletJS API illustrated by the structure of an

The HTML generated by LeafletJS makes clear that all our geographic data is marked up in distinct and nested div elements. As recommended in the HTML Standard class attributes are used to classify these elements in sensible terms. These terms match the terms in the LeafletJS JavaScript API and this way we could relate each DOM Element to the corresponding Leaflet JavaScript class responsible for it. As the HTML tell us, LeafletJS approaches semantic markup using class names in conjunction with the generic div element. Our approach is now to, according to our investigations of the vocabularies, let LeafletJS generate a significantly improved markup instead which turns an annotated web map into one or many machine-readable statements helping tools and users to analyze and interpret the information used to build this geovisualization.

#### 4.1 Annotating elements in the Marker and Popup layers

In the DOM (Document Object Model<sup>9</sup>) of our web map we find all the Overlays we are interested in annotating as child elements of the Objects Pane. For example the standard blue LeafletJS pin marker symbol using the default icon is simply an img element classified as “leaflet-marker-icon” and inserted as a direct child element of the Marker Pane.

To annotate a simple image marker using the Microdata syntax in HTML the map maker or developer needs to pass a valid name of a Schema.org type as an additional parameter to the standard marker API. The only restriction is that the type name given must have a “geo related” property in its type definition<sup>10</sup>. And if the property name to specific the geographic extent of the map element is not named geo but, for example, locationCreated, the property name must be passed into the option called geoprop. Therewith the complete marker element gets automatically written inside a new article element (instead of the original div), which groups and relates the metadata provided (all optional) expressed as meta elements for the marker. The geographic references of the marker are then automatically translated by our plugin and exposed in HTML as a metadata property for the given type, representing a Place with the Geo Coordinate value. Our implementation is not tested with all LeafletJS extensions out in the wild but it is known to handle many standard use-case

<sup>9</sup> Any HTML document can be accessed and manipulated through the DOM, the Document Object Model Interface. The DOM as a “platform and language neutral interface to dynamically access and update the content, structure and style of a document” (W3C 2008) is exposed by the web browser to JavaScript developers such as LeafletJS contributors.

<sup>10</sup> Anything having the item types Place, Geo Shape or Geo Coordinate as direct property in its definition as it is for example the case with every subtype of Place or Person, every type of Organization or Creative Work like Book, Article or Blog but also with Events.



analogous to adding a marker completely opaque to the user. When wanting to annotate Popups the very same logic and handling applies as just described for a Marker.

## 4.2 Annotating Geographical Overlays - Vector Layers

Since our plugin can inspect all LeafletJS objects along with their attributes when they are called and executed we can directly build upon the knowledge built into LeafletJS API when abstracting various geographical file formats and map them to the respective Schema.org types. A geometrical vector layer, like for example a GeoJSON Layer is therefore automatically translated to a statement containing a property of type Geo Shape and not Geo Coordinate. The user of our plugin must not care about mapping LeafletJS terms to the spatial terminology currently defined in Schema.org. The specialty in translating geometrical elements into HTML is that another markup language, namely the SVG (W3C 1998a) standard comes into use and gets directly embedded as a markup fragment into our HTML document. At the moment our plugin implementation is untested with any version of the Microsoft Internet Explorer but is tested with current versions of Chromium from Google and Firefox from Mozilla<sup>11</sup>. To remain standard compliant our plugin writes all semantic annotations into a new metadata element (W3C 1998b) which gets placed within the corresponding g element representing our geometry (or a part of it). For reasons of consistency we rendered simple meta elements W3C (2014a) as content of the metadata element.

```

▼ <div class="leaflet-overlay-pane">
  ▼ <svg class="leaflet-zoom-animat" width="2635" height="918" viewBox="-617 -409 2635 918" style="transform: transli
    ▶ <g>...</g>
    ▶ <g>...</g>
    ▼ <g>
      <path stroke-linejoin="round" stroke-linecap="round" fill-rule="evenodd" stroke="#0033ff" stroke-opacity="0.5"
        clickable"></path>
      ▶ <metadata itemscope itemtype="http://schema.org/Organization" data-internal-leaflet-id="26">...</metadata>
    </g>
  </div>
  ▼ <g>
    <path stroke-linejoin="round" stroke-linecap="round" fill-rule="evenodd" stroke="#0033ff" stroke-opacity="0.5"
      44L726 45L714 47z"></path>
    ▼ <metadata itemscope itemtype="http://schema.org/Country" data-internal-leaflet-id="34">
      <meta itemprop="name" content="Charmingly Inaccurate - United States of America">
      <meta itemprop="description" content="A caricature of the lower 48 United States, this linework set is recogr
        friend you have who's always embellishing his stories. You never mind his lying, though, because the exaggera
      <meta itemprop="url" content="https://github.com/mapsam/project-linework/tree/master/linework-sets/charmingly
      <meta itemprop="http://purl.org/dc/elements/1.1/creator" content="Daniel P. Huffmann">
      <meta itemprop="http://purl.org/dc/elements/1.1/publisher" content="Daniel P. Huffmann">
      <meta itemprop="http://purl.org/dc/elements/1.1/date" content="10/04/2013">
      <meta itemprop="http://purl.org/dc/elements/1.1/rights" content="Public Domain">
      <meta itemprop="http://purl.org/dc/elements/1.1/source" content="Derived from a 1920 broadside map entitled "
        Papers, Acc. 22002, Library of Virginia.">
      <meta itemprop="http://purl.org/dc/terms/created" content="3/24/2013">
      <meta itemprop="http://purl.org/dc/terms/modified" content="10/04/2013">
      ▶ <meta itemprop="geo" itemtype="http://schema.org/GeoShape">...</meta>
    </metadata>
  </g>

```

Figure 2. Annotated GeoJSON Layer rendered in Chromium with SVG.

In the case of annotating a GeoJSON<sup>12</sup> file consisting of a MultiPolygon geometry all polygons described in it are redundantly annotated as entities, e.g. representing the Administrative Area of “French Polynesia”.

Notably no element generated by LeafletJS comprises a unique identifier in the DOM nor does the API allow to set one on the corresponding HTML element. This leaves elements, and in our cases all statements made in geographic web maps non-addressable for others and to prevent this our extension allows map makers to provide a unique domId for their map element so it can become the target of a hyperlink in the WWW. We also discovered that the logical groupings of LeafletJS Markers or Layers are not preserved when the library transforms them into HTML. LeafletJS flattens out all logical groupings possibly arranged in JavaScript into and within their respective container element. A possible solution for this may be to render these logical groupings in an separate but hidden HTML fragment representing the logical groupings of map elements constructed via the LeafletJS API containing references to the elements of the map involved in such groupings. In preparation for this we included a data-leaflet-internal-id attribute on the annotated map elements but the serialization of this groupings was considered out of scope for the herewith documented work. The complete options of the extended standard LeafletJS API can be found at (painted for peer-review).

<sup>11</sup> Versions of Mozilla Firefox and Google Chromium at least supported by our plugin<sup>are</sup> Firefox 45.x and Chromium 49.x

<sup>12</sup> GeoJSON Specification <http://geojson.org/> - As the GeoJSON specification allows to transport various data in properties of each Feature and it would be great to see tools producing GeoJSON files starting to integrate our terminologies proposed here.

## 5 Conclusions

Through the mapping of the spatial terminology from LeafletJS to Schema.org we made it very easy to make essential information contained in every geographic web map accessible, not only to search engines but also to creative software developers interested in working with web maps and HTML.

Following a user centered approach we deduced and selected questions of critical map readers to analyze and combine existing metadata standard terminology for annotating geographic web maps in HTML. We understood geographical web maps, but specifically map mashups, as media and software installations of composite structure. To acknowledge this understanding in our web maps, we presented a way to express (supported by “most major search engines”) multiple authorships in a single HTML document.

Furthermore we've enabled map makers to backup their visual (rhetoric) statements implicit in their geographic web map with potentially more accessible textual (formal) statements about the entities as well as the information and its sources leading to the entities being represented in the map. Through defining what we understand as a geographic web map we could develop a supportive, alternative presentation (textual) for geographically mapped information. Furthermore this model allows me and others to start building visualizations concerned with maps themselves, or with reading of maps. I did so in the assumption that alternative modes of presentation for the information encoded in geographic web maps can assist map readers in analyzing and reflecting the message of a map.

Furthermore I hope that the results of this investigation will lead to a significantly increased inter-operability of information published in form of geographic web maps. This emphasize on the information organization practices behind map making tries to preserve the time humans spent making maps and tries to make the results more accessible. A well described and annotated geographic web map will by definition tell us as readers very much about its context of production and its various authors or contributors.

It is the approach outlined in this, of course quite technical, report that I understand as possible fruitful for advancing questions in cartographic research too – simply through “unfolding mapping practices” (Kitchin, Gleeson, Dodge, 2012, p.1) and for example, focusing on advancing just the authoring or the reading experience for users of interactive geographical web maps.

Building on HTML, SVG and Schema.org the result reported in this paper supports map makers who build on LeafletJS in accomplishing three things:

1. expose essential geographic information implicit in any geographical web map to HTML in a well defined, machine readable way
2. semantically annotating and classifying pinpoint type markers but also geometric overlays and popups in terms of the authors domain of interest (as far as it is yet represented in the Schema.org vocabulary)
3. provide and expose meta data essential for critical map readers to contextualize certain information (as defined in section 2.3) represented through one of the core elements of a so called “map mashup” (Marker, CircleMarker, Popups and GeoJSON Overlay)

The open source licensed LeafletJS mapping libraries enabled us to implement all of the markup refinements we envisioned. Map publishers building on LeafletJS can now install a plugin which makes their geographic web map machine readable while not altering the visual representation of the map formed by various web browsers. The source code of the plugin is available at <https://github.com/mukil/Leaflet.annotate> for installation, adaption, study and distribution and while welcoming any feedback of users to the current implementation we also welcome any contribution leading to its improvement.

In 1992 John Brian Harley published an article about “Deconstructing the map” and in the passage on the “cartographic text” he talks about how he read and can also understand maps as narratives and I think we can easily understand our map mashups as narratives, to especially when multiple authors (actors) contribute certain information into one big complex arrangement. As Harley describes, it often are the “footnotes” or “marginalia” of a map, but especially of early and historical maps, which become essential for being able to interpret and contextualize the information presented in it. Now, if this work is able to equip creators with semantic markup for geographical web maps, critical map readers will have a plethora of footnotes and marginalia available for interpreting a map.

The “first deconstructionist move”, as Harley (1987, p.5) cites Norris, is “to seek out the moments of self-contradiction where a text involuntarily betrays the tension between rhetoric and logic”. Now, if we read and Norris like, “where a map involuntarily betrays the tension between visualization and information”, I relate this

work to Harley and his thoughts on deconstructing the map to a basic concept in computer science. A concept which tells us that we could deal with both, information and representation, separately (following, for example, the definitions of such by Broy 1998). So following this interpretation the visual dimension of a geographic web map could be seen as its rhetoric and the textual, now formalized set of statements about the world made in the map, could be seen as the maps information – or logic so to say. Therefore I say that the first step to develop support for this “analytical move” is to bring semantic markup into the documents we call geographical web maps in HTML, as started here. The next step would be to build on this new model of a map as a kind of composite information storage, a map as a result of sometimes tedious information organization practice, and start to design more meaningful user interactions for this understanding.

## References

- Agafonkin, V. (2016, Last checked: February 2016), LeafletJS Project. URL: <http://leafletjs.com/>
- Bittner, C.; Michel, B. (2013), *Das Dekonstruieren der web2.0 Karte.*, Springer VS, pp. 111–126.
- Broy, M. (1998) *Informatik: Eine grundlegende Einführung. Band 2: Systemstrukturen und Theoretische Informatik (Springer-Lehrbuch) (German Edition).* URL: Springer; 2. Aufl. 1998. Nachdruck 2003 edition (April 15, 2003)
- Brummermann, H. (2015, Last checked: February 2016), Leaflet Web Components. URL: <https://leaflet-extras.github.io/leaflet-map/demo.html>
- Caquard, S. in an interview for OpenCanada by Eva Salinas, The politics of map making (Last checked, 15 December 2015). URL: <https://www.opencanada.org/features/the-politics-of-maps/>
- Dodge, M.; Kitchin, R. (2007), *Rethinking Maps.*, *Progress in Human Geography* 31, p. 331–344.
- Kitchin, R., Gleeson, J., Dodge, M. (2012), *Unfolding mapping practices: a new epistemology for cartography.*, in *Transactions of the Institute of British Geographers*, by Royal Geographical Society (with the Institute of British Geographers)
- DCMI (2012a) *Dublin Core Metadata Element Set.* (Last checked: March 2016), Version 1.1. URL: <http://dublincore.org/documents/dces/>
- DCMI (2012b), *DCMI Metadata Terms.*, (Last checked: March 2016). URL: <http://dublincore.org/documents/dcmi-terms/>
- Gartner, G. (2009), *Web Mapping 2.0.*, Routledge, pp. 68–82.
- Google (2016, Last checked: February 2016), *Promote Your Content with Structured Data.* URL: <https://developers.google.com/structured-data/?rd=1>
- Johnston, P., Powell, A. (2008), *Expressing Dublin Core metadata using HTML/XHTML meta and link elements.*, *Dublin Core Metadata Initiative.* URL: <http://dublincore.org/documents/dc-html/>
- Kitchin, R.; (2010), *Post-representational cartography.* URL: <http://eprints.maynoothuniversity.ie/3846/>
- Kitchin, R.; Gleeson, J., Dodge, M. (2012), *Unfolding mapping practices: a new epistemology for cartography.*, *Transactions of the Institute of British Geographers* 38, Pages 480–496
- MacWright, T. (2013, Last checked: March 2016), *Announcing MapBox.js 1.0 with Leaflet.* URL: <https://www.mapbox.com/blog/mapbox-js-with-leaflet/>
- Mapbox Inc. (2016, Last checked: March 2016), *Mapbox.js Examples.* URL: <https://www.mapbox.com/mapbox.js/example/v1.0.0/>
- Mattern, S.C. (Last checked, 31. January 2016), *Course Material - Critiquing Maps II.* URL: <http://www.wordsinspace.net/wordpress/2013/09/05/critiquing-maps-ii/>
- MediaSmarts (2016, Last checked: March 2016), *Deconstructing Web Pages - 5Ws of Cyberspace.* URL: [http://mediasmarts.ca/sites/mediasmarts/files/lessonplans/lesson\\_deconstructing\\_web\\_pages.pdf](http://mediasmarts.ca/sites/mediasmarts/files/lessonplans/lesson_deconstructing_web_pages.pdf)

- Norris, C. (1987) Derrida. Cambridge, Mass.: Harvard University Press, 1987: 19.), in, Deconstructing the map., Evanston, IL: Program of African Studies, Northwestern University, no. 3, pp. 10-13, 1992., URL: <http://quod.lib.umich.edu/p/passages/4761530.0003.008/--deconstructing-the-map?rgn=main;view=fulltext>
- Richartz, M. (1995), Generik und Dynamik in Hypertext. PhD thesis, Universität Karlsruhe
- Roth, R. E. (2013), Interactive Maps: What we know and what we need to know., Journal of Spatial Information Science 6, 59–115.
- Roth, R. E.; Donohue, R. G.; Sack, C. M. (2014), A Process for Keeping Pace with Evolving Web Mapping Technologies., pp. 25–52.
- Schiewe, J.; Schweer, M. (2013), ‘Trust in the Field of Map Usage’, Kartographische Nachrichten 2/3, 59–65.
- Sporny, M, Longley, D., Kellogg, G., Lanthaler, M., Lindstrm, N. (2010, Last checked: March 2016), JSON-LD Specification. URL: <http://json-ld.org/>
- Turner, A. J. (2006), Introduction to Neogeography., in O’Reilly Short Cuts.
- W3C (1998a, Last checked: March 2016), Metadata - SVG 1.1 (2nd Edition). URL: <https://www.w3.org/TR/SVG/metadata.html>
- W3C (1998b, Last checked: March 2016), VML - the Vector Markup Language. URL: <https://www.w3.org/TR/NOTE-VML>
- W3C (2008, Last checked: March 2016), DOM - Document Object Model. URL: <https://www.w3.org/DOM/>
- W3C (2013, Last checked: March 2016), HTML Microdata. URL: <https://www.w3.org/TR/microdata/#encoding-microdat>
- W3C (2014a, Last checked: March 2016), Document Metadata. URL: <https://www.w3.org/wiki/HTML/Elements/meta>
- W3C (2014b, Last checked: March 2016), HTML5 Article Element. URL: <https://www.w3.org/wiki/HTML/Elements/article>
- W3C (2014c, Last checked: March 2016), HTML5 Blockquote Element. URL: <https://www.w3.org/wiki/HTML/Elements/blockquote>
- W3C (2014d, Last checked: March 2016), HTML5 DIV Element. URL: <https://www.w3.org/wiki/HTML/Elements/div>
- W3C (2014e, Last checked: March 2016), HTML5 Figure Element. URL: <https://www.w3.org/wiki/HTML/Elements/figure>
- W3C (2014f, Last checked: March 2016), HTML5 Footer Element. URL: <https://www.w3.org/wiki/HTML/Elements/footer>
- W3C (2014g, Last checked: March 2016), HTML5 Link Type Author. URL: <https://www.w3.org/TR/html5/links.html#link-type-author>
- W3C (2014h, Last checked: March 2016), HTML5 Q Element. URL: <https://www.w3.org/wiki/HTML/Elements/q>
- W3C (2014i, Last checked: March 2016), HTML5 Section Element. URL: <https://www.w3.org/wiki/HTML/Elements/section>
- W3C (2015a, Last checked: March 2016), RDFa Core 1.1 - Third Edition. URL: <https://www.w3.org/TR/rdfa-core/>
- W3C (2015b, Last checked: March 2016), Schema.org. URL: <http://schema.org/version/2.2/>
- Wikipedia Community (2016, Last checked: March 2016), HTML. URL: <https://en.wikipedia.org/wiki/HTML>