Introducing *Web Mapping: A Workbook for Interactive Cartography and Visualization on the Open Web*

Robert E. Roth[a,*], Carl M Sack[b], Gareth Baldrica-Franklin[a], Yuying Chen[a], Rich Donohue[c], Lily Houtman[a], Tim Prestby[d], Robin Tolochko[e], and Nick Underwood[a]

[a] *University of Wisconsin−Madison, Cartography Lab (USA), reroth@wisc.edu, baldricafran@wisc.edu, chen449@wisc.edu, lhoutman@wisc.edu, nunderwood3@wisc.edu*
[b] *Fond du Lac Tribal & Community College (USA), carl.sack@fdltcc.edu*
[c] *University of Kentucky (USA), rgdonohue@uky.edu*
[d] *Penn State University (USA), tprestby@psu.edu*
[e] *Quizlet, robin.tolochko@gmail.com*

* Corresponding author

Keywords: cartography education, web mapping, interactive maps, user experience design, open source

Introduction and Purpose: *Web Mapping: A Workbook for Interactive Cartography and Visualization* is an online open educational resource introducing the practical skills needed to develop interactive maps and visualizations on the Open Web (Roth et al. 2020): https://github.com/uwcartlab/webmapping. The technological landscape for interactive, online, and mobile mapping and visualization is broad and evolving at a breakneck pace. There are now an increasing number of proprietary web mapping services, such as ArcGIS Online, CARTO Builder, and Mapbox Studio, that simplify the creation of interactive web maps through easy-to-use point-and-click interfaces. However, most online maps today either directly or through proprietary software utilize the Open Web Platform and related set of royalty-free open web standards that can be interpreted by web browsers. Important open web standards include: Hypertext Markup Language (HTML) for structuring web content, Cascading Style Sheets (CSS) for styling web content, the Document Object Model (DOM) and the JavaScript scripting language for storing and manipulating web content in browser, JavaScript Object Notation (JSON) for exchanging data in-browser, and Scalable Vector Graphics (SVG) for rendering visual web content, among others (see Sack 2017 for a review).

Accordingly, cartography students need to learn not just how to *encode* geospatial data following tenets of cartographic design, but increasingly also to *code* a useful and usable interface for exploring the resulting maps. However, the range of technical competencies for developing maps on the Open Web can be daunting for new cartographers. *Web Mapping: A Workbook for Interactive Cartography and Visualization* bridges this gap, teaching designers how to code from the basics of programming and debugging on the Open Web to advanced interactive mapping and visualization techniques using the Leaflet.js (leafletjs.com) and D3.js (d3js.org) web mapping libraries. Here we describe background on pedagogical research projects leading to creation of the workbook, outline the organization and contents of the workbook itself, and finally suggest several instructional use cases for teaching with the workbook.

Background: The cartography curriculum at the University of Wisconsin−Madison dates to 1937 and currently is part of a suite of introductory and advanced courses in design, geographic information systems (GIS), spatial analysis, and geocomputing. The *Web Mapping* workbook is explicitly designed to serve the advanced *Interactive Cartography and Geovisualization* course that focuses on interactive mapping and user experience (UX) design concepts and client-side development with open source technologies: https://geography.wisc.edu/cartography/education/. Other UW courses cover the basics of cartographic representation using proprietary desktop and web-based design tools, as well as courses on server-side and full-stack development; thus, these topics are referenced only tangentially in *Web Mapping*.

Compilation of *Web Mapping* was facilitated through a sequence of pedagogical research projects conducted in the UW Cart Lab. First, we examined *What to teach?* including broad reviews of interactive mapping and UX design concepts as applied to cartography (Roth 2013) and the range of technologies and competencies needed for developing on the Open Web (Roth et al. 2014). The latter fortuitously led us to choose the Leaflet.js and D3.js web mapping libraries for the course, two relatively new technologies at the time that have remained stable and are complementary regarding cartographic design needs. We then examined *How to teach?*, as we quickly realized that the diversity of open web standards and technologies needed a different approach than the "lab handout" used for teaching self-contained technology. This led us to repackage course content into a spiral curriculum, with weekly lessons overlapping prior lessons and foreshadowing future lessons (hence the "spiral") (Sack and Roth 2017). Finally, we examined *Where to teach?* as campus learning management systems (e.g., Canvas) created artificial student barriers for accessing and

manipulating content intended for use on the Open Web. As a result, we migrated the course content onto the GitHub collaborative coding platform (github.com), with students learning the basics of Git versioning control as part of their assignment submission. We published *Web Mapping* as an open educational resource using GitHub as a hosting platform in Summer 2020 and intend to update it continuously with each Spring semester offering of the course at UW.

Organization: The *Web Mapping* workbook comprises 11 *Chapters*, pitched at about 6-8 hours of material to match a single course week, along with an optional *Introduction* chapter outlining interactive mapping and UX design concepts, an optional *About* chapter summarizing the aforementioned pedagogical research projects, and links to the GIS&T Body of Knowledge Cartography & Visualization section (https://gistbok.ucgis.org/) for more information on cartographic design principles and techniques. The 11 core chapters are organized into three *Units*: an introductory unit on workflows and data for the Open Web, including debugging skills, followed by units on spatiotemporal visualization with Leaflet.js and multivariate visualization with D3.js. Chapter 7 focuses on collaborative coding best practices, and so is less relevant when completing *Web Mapping* independently. Each *Chapter* contains 2-4 *Lessons* that form the spiral curriculum, sequenced in the recommended order resulting from the Sack and Roth (2017) study. Lessons are formatted in GitHub markdown and focus heavily on code examples, breaking down complex blocks into their constituent logics. Lessons also include keyword definitions, design guidance, links to documentation, and helpful tips and tricks. Table 1 provides an overview of lessons and topics across *Web Mapping*.

Table 1: *Web Mapping* Lesson Organization and Content

Unit	Chap	Lesson Topics
1	Workflows & Data	
	1	Setting Up Your Workspace: 1a. The HTML Boilerplate, 1b. Web Directory Setup, 1c. GitHub Setup
	2	Scripting and Debugging: 2a. Exploring the DOM, 2b. JavaScript Basics, 2c. jQuery, 2d. Debugging in the Developer Console
	3	Data & AJAX: 3a. (Geo)Web Data Formats, 3b. AJAX Concepts & Syntax, 3c. Understanding AJAX Callback Functions
2	Designing with Leaflet	
	4	Leaflet Foundations & Using Online Resources: 4a. Leaflet Tutorials & API Documentation, 4b. Using Examples, 4c. Using Help Forums, 4d. Finding Tilesets & Data
	5	Dynamic Mapping with Leaflet: 5a. Making Leaflet Layers Dynamic, 5b. Zoom, Pan, & Retrieve Interactions, 5c. Sequence Interaction, 5d. Additional Interaction Operators
	6	The Internal Logic of Leaflet: 6a. Procedural & Object-Oriented JavaScript, 6b. Extending Leaflet Objects, 6c. Using SVG Graphics
	7	Collaborating with GitHub: 7a. GitHub Collaborative Coding Functionality, 7b. Setting Up A Collaborative Repository for a Group Project, 7c. Best Practices for Collaborative Coding
3	Designing with D3	
	8	D3 Foundations: 8a. D3 Selections & Blocks, 8b. D3 Data, 8c. Scales, Axes, & Text
	9	Mapping in D3: 9a. D3 Helper Tools & Techniques, 9b. D3 Projections & Path Generators, 9c. The D3 Graticule Generator
	10	Coordinated Visualizations: 10a. Dynamic Choropleth Symbolization, 10b. Drawing a Coordinated Visualization
	11	Coordinated Interactions: 11a. Dynamic Attribute Selection, 11b. Linked Retrieve Interactions

Instructional Use Cases: *Web Mapping* is freely available for reuse and extension with attribution. *Web Mapping* is optimized for a 15 week semester, with the 11 weekly chapters followed by a four-week group final project. For the lowest engagement, *Unit 2: Designing with Leaflet* (Lessons 4-6 only) can be assigned as an intensive short course or workshop spanning one week. *Unit 1* and *2* can be assigned with a three week final project for a 10 week quarter. Finally, *Unit 3: Designing with D3* can be substituted with a module on spatial databases to cover full stack development. Personal instances of *Web Mapping* for instructional use can be forked and hosted on GitHub. We also provide a utility to convert the GitHub markdown into HTML for copy+pasting into an LMS: https://uwcartlab.github.io/html-to-md/

Acknowledgments: This research was supported by the National Science Foundation CAREER Grant #1555267.

References:

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., R.G. Donohue, C.M. Sack, T.R. Wallace, and T.M.A. Buckingham. 2014. A process for keeping pace with evolving web mapping technologies. *Cartographic Perspectives*. 78: 25-52.

Roth R.E., C.M. Sack, G. Baldrica-Franklin, Y. Chen, R. Donohue, L. Houtman, T. Prestby, R. Tolochko, and N. Underwood. 2020. *Web Mapping: A Workbook for Interactive Cartography and Visualization on the Open Web*. The University of Wisconsin Cartography Lab. DOI: 10.5281/zenodo.4685763

Sack C.M. 2017. Web Mapping. *The Geographic Information Science & Technology Body of Knowledge*. (Ed) John P. Wilson. DOI: 10.22224/gistbok/2017.4.11

Sack C.M. and R.E. Roth. 2017. Design and evaluation of an Open Web Platform cartography lab curriculum. *Journal of Geography in Higher Education*. 47(1): 1-23.