

This is an excerpt from Frank Elavsky's dissertation on *Tool-making as an Intervention on the Accessibility of Interactive Data Experiences*, which can be accessed in full at this archival link:

<http://reports-archive.adm.cs.cmu.edu/anon/hcii/CMU-HCII-26-103.pdf>

This document contains the following sections:

- Abstract
- Table of contents
- Chapter 1: Introduction
- Chapter 2: Background & Related Work
- Chapter 3: Overview of Contributions
- Chapter 5: *Data Navigator*: Low-level Tooling for Creating Navigable Data Structures
- Chapter 9: Discussion & Future Work (Sections 9.1 and 9.3 only)
- References

This document does not contain the following chapters:

- Chapter 4: *Chartability*: Heuristics as a Tool and Resource
- Chapter 6: *Skeleton*: Visual Authoring of Non-visual Data Experiences
- Chapter 7: *Cross-perception*: Rethinking Input Design Towards Blind Analytical Interaction
- Chapter 8: *Software*: Enabling Personalization of Interactive Data Representations for Users with Disabilities
- Chapter 10: Biographical Sketch

Contents

Abstract	v
I Introduction	1
1 Introduction	3
1.1 Is “accessible visualization” really an oxymoron?	3
1.2 On <i>tools</i> , <i>tool-making</i> , and <i>human-tool</i> interaction	5
2 Background & Related Work	7
2.1 Practitioners and Tools	7
2.1.1 Understanding Builders, Makers, Designers, and Developers	7
2.1.2 Approaches to Tool-making in Human-Computer Interaction	7
2.2 Data, Accessibility, and Data <i>and</i> Accessibility	8
2.2.1 Advancements in Interactive Data Visualization and Data Science	8
2.2.2 Accessibility and Assistive Technology in Research versus Practice	9
2.2.3 Data and Accessibility	10
3 Overview of Contributions	13
III Navigation: Making Data Structures Traversable	17
5 <i>Data Navigator</i>: Low-level Tooling for Creating Navigable Data Structures	19
5.1 Abstract	19
5.2 Overview	19
5.3 Related Work	21
5.3.1 Accessibility research and standards in visualization	21
5.3.2 Visualization toolkits and technical work	22
5.3.3 Considering assistive technologies and input devices	24
5.4 Data Navigator: System Design	25
5.4.1 Structure	26
5.4.2 Input	28
5.4.3 Rendering	31
5.5 Case Examples with Data Navigator	33

5.5.1	Augmenting a Static, Raster Visualization	33
5.5.2	Building Data Navigation for a Toolkit Ecosystem	35
5.5.3	Co-designing Novel Data Navigation Prototypes	37
5.6	Limitations and Future Work	40
5.7	Conclusion	41
VI	Conclusion	43
9	Discussion & Future Work	45
9.1	What is a “tool?” A reflection on the social and material identity of tools	45
9.3	Who is responsible for repair?	46
	References	49

List of Figures

5.1	Data Navigator provides data visualization libraries and toolkits with accessible data navigation structures, robust input handling, and flexible semantic rendering capabilities.	20
5.2	Existing accessibility trees and lists, shown using node-edge graph conventions. (*) Denotes only <i>screen reader</i> access. (**) Denotes <i>screen reader</i> , <i>keyboard-only</i> , and <i>pointer</i> access as well.	23
5.3	A. Map of engineers per capita of US states. B. Tree representation of the map data where states are listed alphabetically and also include links to neighboring states. The structure repeats itself if users navigate in a loop. C. Graph representation with the same navigation potential without redundant rendering.	26
5.4	An example of how a single edge instance references a navigation rule and can even have multiple navigation rules. A navigation rule can be referenced by multiple edges.	27
5.5	A generic edge, such as “any-return” can be applied to any node. Function calls handle dynamically assigning the edge’s source and target nodes on-demand. . .	28
5.6	An example navigation rule to move “left” can be called as a method by an event from any input modality. Some examples include common modalities such as touch swiping (A) or speaking “left” (B). This also includes advanced or future modalities such as gesture recognition (C) or touch-activated, fabricated interfaces (D).	29

5.7	An example of how navigation within Data Navigator could use semantic nodes as hyperlinks to provide access to other areas in an application. Alabama has a child node “Counties” which is a semantic HTML link element pointing to a table of counties, outside of Data Navigator’s graph structure. A link is provided to return.	30
5.8	A. The data specified for a node with a reference to separate data that is used to render that node. B. The node will render as a path at the specified Cartesian coordinates. C. This rendered node may then be placed over a visual.	31
5.9	A. A raster (png) visualization of a stacked bar chart showing how 4 English teams performed across 3 major trophy contests. B. An example navigation schema that allows children nodes to have 2 parents (two tree structures intersecting), one for contests and one for teams. C. An example of Data Navigator’s navigation logic abstraction, which allows edge types to have programmatic sources, targets, and rules, such as a single rule that gives all nodes a edge to exit the visualization. D. An instantiation of the schema, showing all corresponding rendered nodes and their edge types according to the schema design and navigation rules.	32
5.10	A. Various charts from <i>Vega-Lite</i> share the same general structures with each other when rendered using canvas (B) or SVG (C). D. With Data Navigator, we replicated the existing SVG navigation pattern (C) but used a canvas-based rendering for the visualization. E. We also improved the navigation scheme to nest marks within a mark group to allow users to skip them, if needed.	35
5.11	Our material preparation process involved taking a reference (A), tracing it (B), and rendering it on a tactile display (C).	37
5.12	A. A reference image from <i>Penrose</i> of a set diagram containing two sets intersecting. B. A diagram of our proposed structure, with three levels of information.	38
5.13	A. A reference image from <i>Penrose</i> of a parallel vectors diagram. B. A diagram of our proposed structure, with two main sub-categories of information: understanding the vectors and their parallels.	39

Abstract

In this dissertation, we contribute practical advancements in tool-making as an intervention on the accessibility of interactive data experiences. The thesis of this dissertation is as follows: *This dissertation argues that the tools practitioners use to build interactive data experiences are themselves sites where accessibility barriers are produced, prevented, or alleviated for both end users and authors. This work contributes five tools—Chartability, Data Navigator, Softerware, Cross-perception, and Skeleton—that collectively center accessibility work on the empowerment of disabled and non-disabled practitioners across the full arc of evaluation, data navigation, analytical interaction, and personalization.*

Rather than framing accessibility research solely around ideal experiences for end users with disabilities, this thesis investigates why accessibility work is so difficult for the practitioners who build interactive data experiences and what tool-making can reveal about those difficulties. We organize this investigation around four domains where practitioners face the most persistent challenges: evaluation, navigation, interaction, and personalization. *Chartability*, a heuristic framework contributed first and maps the full landscape of accessibility barriers and identified these three latter domains (navigation, interaction, and personalization) as the areas where the most severe gaps remain. *Data Navigator* and *Skeleton* then investigate navigation, finding that practitioners struggle because navigation structure has no visible, manipulable representation in their workflows. *Cross-perception* engages interaction, demonstrating that blind data analysis has been constrained by existing tools and that a new interaction design framework can reshape what analytical work is possible. *Softerware* addresses personalization, revealing that access needs genuinely conflict across users and that meaningful personalization requires system-level infrastructure that does not yet exist in practitioner tooling ecosystems.

Combined, these contributions provide empirical insights and practical advancements in the state of the art for tooling that bridges gaps in current accessibility practices in visualization and data science. Our work ultimately enables people with and without disabilities to better evaluate barriers in, analyze with, design for, develop, and personalize interactive data experiences. We demonstrate that tool-making is a productive intervention that both engages accessibility barriers and elucidates why those gaps exist in practitioner work.

Part I

Introduction

Chapter 1

Introduction

This thesis is a body of research situated within the existing research area focused on making interactive data representations more accessible for people with disabilities. Much of the work in this existing area is situated within the context of making interactive data *visualizations* accessible, particularly (but not exclusively) for people who are blind. My work, contributed here in this thesis, is focused on using *tools* as a specific intervention and sub-area of study for making interactive data *representations* accessible for people with disabilities, broadly speaking. (“Representations” here is an intentionally broader term than “visualizations,” which are exclusively visual representations of data.)

Before we begin, two things must be understood up front, or else the rest of this thesis could be interpreted with disruptive assumptions: we must interrogate the phrase “making visualizations accessible” and unpack why *tools* are a meaningful area of study.

1.1 Is “accessible visualization” really an oxymoron?

The first assumption that must be disrupted is perhaps the motivating cornerstone of this research, which is that the phrase “making visualizations accessible,” while a noble goal, is not the semantically correct phrasing nor precisely what describes my work. This can be misleading. I do use the phrase “accessible visualization” but will admit that this seems to confuse certain people with very particular opinions about things. We will clear this up.

Villains of our field’s past have written incendiary and ableist perspectives on why “no forms of data visualization, not just dashboards jam-packed with graphics, can be made fully accessible to someone who is blind,” and that “[a blind man] will never be able to analyze data as I do visually, because many aspects of vision cannot be duplicated by his other senses” [28]. However, this position misunderstands what the goal of accessibility is, and arguably even what the goal of visualization itself is.

Making visualizations accessible *isn’t* about the visualization, it’s about making the outcomes of the visualization accessible.

Visualizations are ubiquitous and paramount for decision-making. However, the *artifact* that is a visualization is not even the goal of the act of visualizing: developing understanding, insight, confidence, and communication among and between human beings are the goals of visualization. Visualization is about making data easier to use for all kinds of things. Yes, our visual system enables us more than any other form of sensory cognition that we have [10, 33, 113]. But we aren’t trying to make sight itself accessible. We are trying to make it possible for people to make meaningful decisions, gain valuable information, build conjectures, and effectively communicate with others.

Many, many people who I’ve spoken to over the course of my career, even before embarking on this thesis journey, misunderstand this simple fact: making a “visualization” accessible *isn’t* about the visualization itself but rather making what the visualization is meant to *accomplish*

accessible. It's about equal outcomes, not equal interactions with an artifact.

People with disabilities are no small portion of the world's population. In the United States, 27% of people self-report living with at least one disability that affects their daily lives [84] and all of us will eventually age into disability (if we are lucky to live a long life).

People with disabilities (again, that will be all of us *eventually*) deserve to participate fully in life. They deserve financial independence. They also deserve loving care and interdependence. People with disabilities have a right to make informed decisions, to know about the status of a global pandemic, and to have an understanding of local and national politics [27]. While we use visualizations to navigate all of these domains, the goal is not to make the charts and graphs themselves somehow equally useful to all people. That would be a false measurement of success.

Our goal then, is measured by the success of lives led by people with disabilities [116]. Many other measurements are just metrics along the journey towards that goal. We then ask: Can people with disabilities also use data to live full lives? Can they make *fast* decisions based on data? Meaningful, careful, *slow* decisions? Communicate complex ideas? Crunch and clean data, develop models, find errors, and build hypotheses? Can they have memorable, immersive, beautiful, aesthetic experiences with data too [53]? Making "visualizations" accessible really is a misnomer. We are ultimately trying to make everything about what interactive data experiences *accomplish* for people equitable and accessible.

Again, if the goal of accessible visualization were about visualizations themselves, then the correct course of action would be one framed by the medical model [77]: that there is a normative state of behavior and capability (in this case, it would be "normal" to be able to read a visualization and make a decision) and any deviation from that norm must be corrected. This framing first assumes that the visualization should not be altered or improved. And then this framing puts the burden on the bodies of people with disabilities: that they must be "fixed" and given sight or brought to some equivalent state as someone who is "healthy," normal, and sighted. Plenty of scholars have already discussed why this framing is a problem, not only because it places undue burden on people with disabilities, produces pathologies and hierarchies of disability, but also because it is fundamentally not economically or ethically feasible.

So we then turn to other models of disability, such as the social model. The social model is heavily discussed by disability scholars and is not the end-game or last and total way of thinking about disability [77, 88, 90, 111, 139]. But the core motivation is that society, not medicine, is also a path towards solving problems that people with disabilities face. A few important concepts and concretely actionable things come from the social model that can help motivate the work of this thesis.

First, we look to the historical birth of the social model of disability: in the 504 sit-ins that took place in the United States in 1977. Cities had curbs and curbs are a barrier for people who use wheelchairs. So protests happened because decisions were being made without people with disabilities at the table. In this instance, people acknowledged that political power was an exclusive club and fought to ensure their cry "nothing about us, without us!" materialized.

And this leads us to the first and most-foundational philosophical framing for this thesis: that our *artifacts*, these things we've created from curbs to data visualizations, can become *barriers* for people with disabilities. And it is then the artifact, not the body of the person with a disability, where disability is produced in this model. Rather than a comparison to a normative state as a way to frame disability (the medical model), we instead must observe and evaluate material

outcomes based on human-made problems.

So, the social model is framed around society “solving” inequities: we get involved and make political and legal change tangible. But a second model also emerges from within the social model: one where we can now frame *who is first responsible* for repair: the curb designers and implementers.

And knowing who is first responsible for access leads us into the moral and ethical imperative that motivates this thesis: the builders and makers of visualizations are ultimately the ones who provide exclusive value for only a subset of people: those *without* disabilities. **We must first change how builders and makers do their work.**

So the phrase “accessible visualization” is really about recognizing that visualizations produce barriers for people. That means that it is our ethical imperative, as builders and makers, to fix them. And that act of fixing barriers leads us away from mere visual representations of data into a wide variety of other senses and interaction modalities. There are many paths forward towards fuller and more-equitable lives led by people with disabilities.

1.2 On *tools*, *tool-making*, and *human-tool* interaction

Then the act of making becomes immensely important: we, the builders and makers of our world, need to get things right; there is a risk involved when making things that we will exclude people with disabilities. We need to make sure that we build a better world than the one we have now. We must care for new things we create and tend to the repair and maintenance of what we’ve already made. And this ethical imperative leads us to the topic of *tools* and *tool-making*.

So the second thing that must be understood before we embark on this thesis is that *tools* are not the same as *solutions* or *applications*. Sometimes tools can be used to *solve* things and are certainly, in ideal circumstances, *applied* in various contexts. But understanding the role of the “tool” in human-tool interaction is paramount for engaging in the work of making anything accessible for people with disabilities.

We use tools to shape our world, break old things, and make new things. But a tool, like the hammer (as an example), does not inherently *solve* something like homelessness. But a hammer can be used to build homes if there are social policies in place and proper resources allocated. This means that for the success of tooling, there is often a larger material, social, legal, and policy reality that supports and necessitates those tools. This thesis will not be focusing on changing the upstream dependencies, but optimistically operating as if they were true (or will be true in time).

However, in some cases, tools can *destroy*. The hammer has a claw and can easily pry apart boards and tear down homes. So tools carry potential to do all kinds of things, both good and bad, and how a tool is used is often open-ended, variable, and heavily dependent on socio-technical realities. Tools participate in personal and political agendas [131] and are sometimes, for this reason, regulated or made proprietary and controlled by powerful entities [36, 129].

So tools are not without any sort of ethics. We cannot just blame tool-users for outcomes when much of a tool depends on these larger systems and structures. Technologies (tools included) encode the assumptions and biases of their *creators* as much as, if not more than, their users. Tools that build things for others to use can be loaded with assumptions about what peo-

ple are *able* to do [132] and also rules and guardrails about what anyone downstream from that tool’s design *should* do [36, 128]. These assumptions, biases, and rules *limit*, *enforce*, *magnify*, *exclude*, and *enable* what a tool-user is capable of.

Tools for visualizing data are a perfect case study in this problem: virtually every major data visualization library, application, or software ever made was made entirely with the assumption that data should be transformed into visual representations. This is a reasonable assumption, since virtually all of the tool-makers are sighted and visualization is incredibly helpful to our cognition of and communication with data [30].

So data visualization, as a field, has focused its tool-making efforts on reducing the difficulty involved in visualizing data. Some visualization tools are concise [96], others are lower level but much more expressive [7]. Tool-making in visualization has focused on making it easier to scaffold a wide variety of interactions both with the visualizations as well as with their underlying data models [44].

However, as time has moved on, people began to speak out about color-vision deficiency in data visualization. Some people, primarily those with X/Y chromosomes (largely men) who are of European ancestry, have a deficiency in their ability to perceive certain colors. Then a plethora of research arose that began to look into the barriers that folks who are colorblind face in data visualization. As a result, our practices and tools improved. We began to educate practitioners, develop new color palettes, researched new methods for testing our designs, and built new systems for handling automatic color encoding. Our tools evolved.

But now data visualizations have arguably become ubiquitous in daily life. By comparison, we have far more tools now for making visualizations quickly and easily than we do for representing data in non-visual ways. We also have far more research, relatively speaking, into how sighted end users interact with visualizations.

So this thesis engages gaps that arise in this space: Practitioners face immense challenges when crafting accessible data experiences. We first need to educate practitioners on what accessibility barriers actually are in interactive visualizations. Then, we must help them engage the hardest barriers in this work and create building blocks that help them to construct navigable data experiences, build design frameworks that can inform entirely new kinds of data interaction, and develop software systems for end-user personalization and agency. Our research seeks to advance the state of the art in tools that assist in accessible data interaction while also using tool-making as an intervention that helps us to better understand and characterize *why* and *how* data practitioners face barriers themselves in this work.

Chapter 2

Background & Related Work

2.1 Practitioners and Tools

2.1.1 Understanding Builders, Makers, Designers, and Developers

Research investigating the practices and experiences of individuals who create with computers employs a range of high-level methods. Ethnographic studies, case studies, and design ethnographies are common approaches, allowing researchers to immerse themselves in communities such as the DIY/maker and assistive technology spaces [52, 55]. These methods capture the nuanced challenges practitioners face when engaging in new and unfamiliar work, including the transition from traditional to digital fabrication, coding, and tool creation [54]. By observing and interviewing practitioners in naturalistic settings, researchers uncover the social, cultural, and technical factors that shape how makers adapt and evolve their work practices.

Participatory design and co-creation are also central to this field [112]. These approaches encourage collaboration between researchers and practitioners or end-users, enabling a deeper understanding of the cognitive and creative processes behind design and development [40]. Such collaborative sessions reveal how designers shift their thinking when encountering novel challenges, embracing iterative processes that blend experimentation with reflection. Similarly, developers often modify their applications, tools, and even programming languages through feedback loops and community-driven innovation, highlighting a dynamic interplay between individual creativity and collective knowledge.

Additionally, design-based and case-study research methods explore how new practices can augment the existing work of practitioners [14, 57]. This involves not merely filling gaps or solving isolated problems but reimagining the possibilities for creative and technical expression. Researchers in this space envision systems that support continuous learning, adaptation, and innovation [38]. The focus is on enabling practitioners to extend their capabilities—providing scaffolds for experimentation, fostering environments where unconventional approaches are encouraged, and integrating new technologies in ways that amplify creativity and intelligence rather than simply addressing deficits [61, 118, 132].

Overall, the research methods used in this area are multidisciplinary, combining qualitative insights with iterative design practices to offer a holistic picture of the challenges and opportunities of builders, makers, designers, developers.

2.1.2 Approaches to Tool-making in Human-Computer Interaction

In human-computer interaction, tool-making research spans both the creation of entirely new capabilities and the enhancement of existing systems. One prominent approach involves piggy-backing on current systems—leveraging their established functionalities to introduce improvements that streamline workflow or unlock new interactions [37]. This method often focuses on integrating with widely used platforms to amplify their usability, enabling users to perform

tasks in more intuitive or efficient ways. By building on existing infrastructures, researchers can demonstrate how small, targeted modifications have the potential to transform user experiences.

Another significant approach centers on the notion of appropriation [18, 21, 93, 115]. Here, research examines how users adapt tools for uses beyond their original intent. Studies in this vein explore the creative processes behind such re-purposing, uncovering the latent functionalities and opportunities that emerge when practitioners modify systems to suit their unique needs. This perspective often leads to the development of modular, extensible tools that encourage experimentation and user customization, fostering a more personalized interaction with technology. In some cases, theory has been developed from the study of emergent and generative tool-use [2, 4], broadly informing future tooling projects as well as general theories of creative human interaction with technology.

Beyond these, tool-making in HCI also includes the development of systems designed to empower users by providing entirely new capabilities, sometimes explicitly named “toolkits” and other times generally just referred to for their ability to enable novel interaction and outcomes [65, 86, 94, 105, 105]. These projects may range from novel software environments that facilitate rapid prototyping and live programming to innovative hardware devices that bridge the gap between digital and physical interactions [46, 48, 86]. The emphasis is not solely on problem-solving but on enabling creative exploration, new possibilities, and even hacking the potential of technologies towards new ends [47]. Such projects often present their contributions through demonstrative prototypes and case studies that reveal potential applications, even if they are accompanied by minimal formal evaluations [25].

This body of work reflects a balance between novelty and practicality. While some projects aim to introduce groundbreaking new ways to interact with data and systems, others refine existing practices to improve efficiency and accessibility. Together, these approaches underscore a commitment to enhancing human capabilities, allowing users to not only solve problems more effectively but also to unlock new avenues for creativity and innovation.

2.2 Data, Accessibility, and Data *and* Accessibility

2.2.1 Advancements in Interactive Data Visualization and Data Science

Recent years have witnessed significant advancements in interactive data science and visualization, driven by innovations that enhance both the performance and usability of data tools. Cross-filtering, as a subtype of cross-linked interaction, has emerged as a powerful technique, enabling users to interact with multiple data dimensions simultaneously [3, 43, 68, 125]. By linking various filters, analysts can quickly build hypotheses and isolate patterns, trends, and anomalies in complex datasets, leading to more informed decision-making. Stress has been placed in recent years on developing fast systems that are optimized showing more and more data at once while reducing latency in user interaction as much as possible [44, 68, 135].

Automated data processing and cleaning have revolutionized workflows by reducing the time spent on manual data wrangling [26]. Sophisticated algorithms now automatically detect inconsistencies, fill missing values, and transform raw data into usable formats. These improvements enable researchers and practitioners to focus more on analysis rather than preparation.

Faster tooling has further accelerated data exploration. Enhanced computational frameworks and optimized libraries allow for real-time data manipulation, making interactive visualization more responsive [44]. Coupled with easier-to-use grammars and scripting languages, these tools lower the barrier to entry, empowering users with limited visualization, geometry, trigonometry, data, and graphics coding experience to generate complex, interactive, visual representations of data [96]. New visualization types and techniques—ranging from dynamic dashboards, faceting, to immersive 3D visualizations—offer novel ways to explore and interpret data [135].

Despite significant breakthroughs, current advancements have largely neglected the needs of people with disabilities. Innovations in data science and visualization have focused on sighted user populations, prioritizing visual clarity and interaction speed using direct pointer techniques (such as with touch or a mouse) [78]. This focus often overlooks accessibility requirements for individuals who are blind, have low vision, experience cognitive or vestibular challenges, or possess motor disabilities that limit traditional pointer use [130].

2.2.2 Accessibility and Assistive Technology in Research versus Practice

2.2.2.1 Research: Focus on Blindness and Computer Output

Research and standards are both somewhat limited by a strong bias towards visual disabilities. In *Chartability*, 36 of the 50 criteria related to accessible visualization considerations involve visual disabilities [24, 27]. Marriott et al. also found that visual disability considerations are the primary focus of data visualization literature [78], leaving the barriers that many other demographics face unstudied. Accessibility research broadly has traditionally concentrated on the experiences of individuals who are blind, investigating how they perceive and interpret computational output [73]. Studies in this area explore alternative modalities for conveying data, such as auditory representations (through synthesized speech), tactile interfaces, and sonification techniques. Researchers focus on identifying effective methods for transforming visual data into formats that blind users can easily comprehend. This body of work not only examines the perceptual challenges but also delves into cognitive processing differences, aiming to optimize the accessibility of complex information and interactive systems for users with visual impairments.

While research has made strides in converting visual outputs into auditory or tactile forms for blind users, interactive input methods remain underdeveloped. Most efforts have concentrated on optimizing screen reader navigation and information retrieval, leaving text entry and command execution cumbersome. Screen readers, as they currently exist, offer limited support for efficient input, making it challenging for users to perform complex interactions. Although tactile interfaces hold promise for providing more intuitive input methods, they are still in the experimental stage and have not been fully integrated into mainstream accessible computing solutions, perpetuating a critical gap in effective user interaction.

2.2.2.2 Practice: Focus on Standards and Specialization

In contrast, practical accessibility efforts are often centered on the implementation and adherence to established standards and guidelines, such as WCAG [120]. There has been a growing interest in developing guidelines for practitioners [22, 24] and even applying guidelines as a method of

validation alongside human studies evaluations and co-design [27, 70, 71, 141]. Existing accessibility standards bodies like the Web Content Accessibility Guidelines do stress the importance of accurate, functional semantics in order for screen reader users to know how to interact with elements [122]. For interactive visualizations this means that button-like or link-like behavior should expressly be made using elements that are semantically buttons and links.

Accessibility professionals, who typically possess specialized expertise, act as intermediaries between the design and development of digital products and the strict requirements of accessibility standards. Their role involves translating abstract guidelines into concrete design solutions, ensuring that websites, applications, and services meet regulatory benchmarks. By focusing on a standards-based approach, practitioners help organizations navigate the complexities of legal and technical requirements, thus ensuring that accessible design principles are integrated into mainstream technology development. This dual focus on rigorous standards and specialized expertise ensures that accessibility is both technically sound and legally compliant across diverse digital environments.

However, accessibility standards are inherently reactive, often lagging behind rapid technological advancements by five, ten, or even twenty years (or more). This delay occurs because developing, vetting, and formalizing standards requires consensus among diverse stakeholders and extensive testing to ensure compatibility and compliance. In contrast, cutting-edge interfaces and computational capabilities evolve swiftly, driven by dynamic market forces and user innovations. Consequently, accessibility guidelines tend to reflect outdated technologies, creating a persistent gap between modern interactive systems and current best practices in accessibility.

2.2.3 Data and Accessibility

In parallel to Mack et al.’s “What do we mean by Accessibility Research?” [73] nearly all topics of study at the intersection of accessibility and data are focused on visualization and vision-related disabilities [130]. Largely, access issues other than vision that affect data visualization (such as cognitive/neurological, vestibular, and motor concerns) are almost entirely unserved in this research space. Kim et al. found that 56 papers have been published between 1999 and 2020 that focus on vision-related accessibility (not including color vision deficiency), with only 3 being published at a visualization venue (and only recently since 2018) [64]. Marriott et al. found that there is no research at all that engages motor accessibility [78]. We have found 2 papers that engage cognitive/neurological disability in visualization and 1 student poster from IEEE Vis, which are all recent (specifically intellectual developmental disabilities [137] and seizure risk [109, 110]). We found no papers that engage vestibular accessibility, such as motion and animation-related accessibility. We also found that there is no research specific to low vision disabilities (not blindness or color vision deficiency) unless conflated with screen reader usage in data visualization. Blind and low vision people are often researched together, but in practice may use different assistive technologies (such as magnifiers and contrast enhancers) and have different interaction practices (such as a combination of sight, magnification, and screen reader use) [114].

Since the 1990s, the most prominent and active accessibility topic in data has been color vision deficiency in data visualization [12, 66, 79, 83, 85]. Research projects that explore tactile sensory substitutions to charts have been a topic in computational sciences dating back to the

1983 [99], with tactile sensory substitutions being used for maps and charts as far back as the 1830s [39]. Sonification used both in comparison to and alongside visualization and tactile methods for accessibility dates as far back as 1985 [9, 15, 29, 76, 80, 140]. Some more recent work has explored robust screen reader data interaction techniques [34, 108], screen reader user experiences with digital, 2-D spatial representations, including data visualizations [97, 102], dug deeper into the semantic layers of effective chart descriptions [70], and investigated how to better understand the role of sensory substitution [13]. Jung et al. offer guidance that expands beyond commonly cited literature that chart descriptions are preferably between 2 and 8 sentences long, written in plain language, and with consideration for the order of information and navigation [60].

A wide array of emerging research projects investigate screen reader users needs, barriers, and preferences, and offer guidelines, models, and considerations for creating accessible data visualizations [13, 27, 70, 102]. Jung et al. offer guidance to consider the order of information in textual descriptions and during navigation [60]. Kim et al. collected screen reader users' questions when interacting with data visualizations, which could open the door for more natural language data interaction [63].

Data visualization accessibility has come far in recent years. But little work has been done to explore what disability scholars call "access friction" - a tension that arises when access must be negotiated [41, 53]. This friction is often a result of static barriers in shared spaces: one artifact or approach designed to include some people may end up excluding others.

Yet despite these resources, making data visualizations more accessible remains a difficult task for practitioners [59, 104]. Some accessibility guidelines even conflict, for example on the topic of patterns and textures used in charts. One side stresses that patterns are harmful to cognitive and visual accessibility [98] while another stresses that redundant encoding strategies are necessary [24].

These difficulties point to a deeper problem: the tools practitioners use to build data experiences were not designed with accessibility in mind, and the resulting gaps are not evenly distributed. Even in an ideal state where guidelines agree, the hardest remaining challenges cluster in three domains: navigation, interaction, and personalization. In these, practitioners lack the structural, technical, and infrastructural means to reason about accessible design and act on those considerations.

Chapter 3

Overview of Contributions

In this dissertation, we contribute practical advancements in tool-making as an intervention on the accessibility of interactive data experiences. The thesis of this dissertation is as follows: *This dissertation argues that the tools practitioners use to build interactive data experiences are themselves sites where accessibility barriers are produced, prevented, or alleviated for both end users and authors. This work contributes five tools—Chartability, Data Navigator, Softerware, Cross-perception, and Skeleton—that collectively center accessibility work on the empowerment of disabled and non-disabled practitioners across the full arc of evaluation, data navigation, analytical interaction, and personalization.*

Rather than framing accessibility research solely around ideal experiences for end users with disabilities, this thesis investigates why accessibility work is so difficult for the practitioners who build interactive data experiences and what tool-making can reveal about those difficulties. We organize this investigation around four domains where practitioners face the most persistent challenges: evaluation, navigation, interaction, and personalization. *Chartability*, a heuristic framework contributed first and maps the full landscape of accessibility barriers and identified these three latter domains (navigation, interaction, and personalization) as the areas where the most severe gaps remain. *Data Navigator* and *Skeleton* then investigate navigation, finding that practitioners struggle because navigation structure has no visible, manipulable representation in their workflows. *Cross-perception* engages interaction, demonstrating that blind data analysis has been constrained by existing tools and that a new interaction design framework can reshape what analytical work is possible. *Softerware* addresses personalization, revealing that access needs genuinely conflict across users and that meaningful personalization requires system-level infrastructure that does not yet exist in practitioner tooling ecosystems.

We engage each domain below with the questions: “Why does this work matter?”, “Why is it hard?”, and “What has tool-making within this domain showed us?”

The 4 domains of work that I engage in this thesis start first with **evaluation**. In work contexts where someone is designing and developing interactive data experiences, the practitioner must have the knowledge, tools, and resources available to systematically identify how their interfaces produce barriers for people with disabilities. A significant portion of professional accessibility work (arguably most, if not all) is founded on auditing and evaluating barriers to access. This work is pre-dominantly done through a standards-based approach [120], although in more robust evaluation work, people with disabilities are actively involved in the process [87].

Evaluation is difficult work because much of it is contextually defined by the author themselves, and most tasks at this intersection require careful, non-automated processes and methods [17, 87]. To make matters more difficult, no comprehensive guidelines, tests, and tools exist in any singular location. Practitioners often must gather these resources themselves, which tend to be situated towards accessibility in general or are high-level and provide minimal usefulness in practice. Additionally, practitioners themselves often have little knowledge about the veracity or quality of any given bit of information they gather [59, 104], and often do the work themselves to synthesize this disparate space of information into a usable format they can apply to their own

evaluation work.

To engage this, our first main chapter focuses on *Chartability*, a heuristic framework that enables designers, developers, and auditors to systematically evaluate data visualizations and interfaces for a wide range of accessibility barriers, considering people with visual, motor, vestibular, neurological, and cognitive disabilities. In this project, we did the hard work for other practitioners and contributed our collection of synthesized accessibility resources in a single workbook. We had practitioners try out our resource in real environments, in-situ, in order to learn more about the challenges and barriers they themselves faced in evaluation work. With *Chartability*, practitioners, especially those with limited accessibility expertise, gained more confidence and clarity in assessing and improving their work. Additionally, *Chartability* has since become widely applied as a framework that isn't just used for evaluation but also as design guidance in many contexts, internationally, including policy organizations, governmental groups, and more than 100 companies and businesses.

Chartability then opened up a significant landscape of new projects and research directions. From a combination of my existing expertise as a visualization designer and engineer, in addition to continued application of *Chartability* in the wild, we began to identify the trickiest and most-difficult domains of work for practitioners. *Chartability* has 50 total heuristics, or tests, each organized under one of 7 principles. But 3 larger domains began to emerge as the areas where the most severe and dramatic accessibility barriers remained unaddressed: on data *navigation*, analytical *interaction*, and interface *personalization*.

So the next section of this thesis engages the first of these three: **navigation**. Navigation is a fundamental type of interaction that is leveraged by modern software-based assistive technologies. Screen readers, the primary tool used by people who are blind to interact with computers, navigate content. Additionally, many other assistive technologies, such as a sip and puff device (like the "POSSUM" from as far back as '63 [75]) also navigate. Navigational technologies are leveraged by people with a significant array of disabilities, yet tend to be entirely ignored by existing data visualization tools, which are pre-dominantly built to support direct input (using a computer mouse).

Empirical work has already demonstrated that structural navigation is actually good [141], even when regular alternative text (image descriptions) exist. This is both because people who are blind can gain both a high level understanding (from the description) as well as lower-level sense of the data's structure and arrangement, in addition to the fact that discrete, structural navigation exposes interactivity that may exist on any visualization elements (such as they can be hovered or clicked with a mouse in order to perform some action). So, if good empirical work exists: *why haven't practitioners put this research into action? What makes this work hard to do?*

We first built *Data Navigator* to provide the building blocks we needed in order to address the technical and conceptual gaps that were required to make any visualization or visualization tool provide a navigable, interactive structure. *Data Navigator* is a low-level toolkit which can be used to construct accessible navigation structures such as lists, trees, and diagrams from an underlying graph structure. We leveraged graph theory for an applied HCI problem: nodes and edges represent any relationships within the data as a structure, which then supports rich expressiveness of data navigation experiences. Users can navigate discrete marks in a visualization, clusters, groupings, and more. In addition to its structural scaffolding, *Data Navigator* also supports a wide array of input modalities leveraged by people with disabilities (screen readers,

keyboards, speech, and gestures).

Data Navigator provided a substrate, but this contribution alone wasn’t enough to engage the question *why is navigation so hard, in practice?*. So the chapter following *Data Navigator* introduces *Skeleton*, a data navigation authoring tool built on top of *Data Navigator*. Our novel approach in *Skeleton* involves visualizing and making manipulable the nodes, edges, and textual data that comprise non-visual end user experiences. *Skeleton* visualizes the building blocks that comprise *Data Navigator*. Additionally, *Skeleton* provides expressive, rapid scaffolding capabilities that leverage data visualization rendering engines. This scaffolding engine helps practitioners quickly create common configurations for non-visual data navigation structures that retain visual congruence to the underlying structure.

But most importantly, *Skeleton* serves as a framework that shapes designerly consideration. Our conjecture was that because sighted practitioners cannot *see* navigation building blocks, they will not treat those elements as iterable design materials. We conjectured: Navigation is hard in practice because sighted designers face barriers to iteration and understanding. We conducted an empirical study with sighted practitioners and found that making non-visual elements visual helped practitioners shift from treating accessibility as a compliance task to treating it as a design problem, re-iterating on the visual aspects of their design, and engaging in the complex and nuanced components that comprise data navigation experiences.

Now, **interaction** becomes the next area we wanted to engage. Existing accessible data interaction for people who are blind, including our previous work on navigation (which is a form of interaction), predominantly seeks to expose information. This is what we call *access-oriented interaction*. In terms of low-level analytical tasks, most are then made feasible through navigation, sonification, or summarization-based and question-answering approaches: retrieving values, filtering, computing derived values, sorting, determining ranges, clustering, and finding outliers. What remains are analytical tasks that, despite being “low level” (understood as *unable to be reduced into more fundamental tasks*), are cognitively highly complex: finding correlations and characterizing distributions [1]. These tasks require complex hypothesization and exploration, rather than a system that simply encourages surfacing what is known or what is already present in the data: it requires combining, remixing, restructuring, and dividing data.

But blind *analytical interaction* isn’t just important to engage because it is understudied, it is important to engage because many of interactive information visualization’s most impactful tools for data science enable it [29, 44, 82, 125]. In visualization, *cross-filtering* is one example of an interaction that enables a user to filter one visual space while seeing a coordinated change in another visual space simultaneously and near-instantaneously. The speed of input interaction and perception of output also matters: even a small bit of latency changes the quality of a user’s data exploration activities [68]. We conjectured that a screen reader, the most-used tool leveraged by blind people when interacting with computers, may be insufficient for engaging this task.

To engage this, this thesis introduces *Cross-perception*, an approach for building analytical interactions that support perception in one space of input interaction with simultaneous, non-competing perception of output in another space of data representation. We first formalized a design framework for producing *cross-perception* experiences and then built a novel prototype device, the *cross-feelter*, that enables blind *cross-perception* of a cross-filtering data exploration interface. In an empirical study with blind users (with and without existing data expertise), we found *cross-perception* speeds up analytical exploration by 90% and helps blind users consider

vastly more questions of their dataset (+188% computational queries, +54% spoken aloud) compared to a screen reader-driven interaction.

Beyond performance, we found that our input modality itself shaped the character of analytical engagement: participants didn’t just work faster, they considered more dimensions of their data and asked qualitatively different questions. The *cross-feelter* also reduced anxiety and substantially increased enjoyment, particularly for participants without prior data expertise, suggesting that the barriers blind practitioners face in data work are not only functional but affective. Additionally, we had our blind practitioners imagine new interaction possibilities that *cross-perception* could enable including and beyond our *cross-feelter* device.

Our final domain of work is the most difficult for visualization practitioners to engage: **personalization**. While *navigation* demanded better software tooling and visual support and *interaction* required new hardware, *personalization* completely re-orientes how software authoring takes place. Personalization matters because of *access friction*, which is a design challenge where one design or interface configuration that might be accessible for one person or group of people turns out to create barriers for someone else [41, 53]. In existing work on personalization and accessibility, studies have demonstrated that end-user control is great to have and can alleviate friction [58, 136], but little work has been done to explore what personalization looks like for an existing data visualization library and how practitioners should build and maintain their existing systems to support it.

Our final chapter introduces *Software* to address the tension between standardized accessible design and the diverse needs of real users with disabilities. In the wild, *access friction* exists in every design that reaches a public audience; it is inevitable. This tension ultimately means that some users have a worse experience, and may even face exclusion, with any particular design configuration. So for this work, we conducted our research in-situ with visualization software engineers and designers and worked to build a scalable, flexible software system dubbed “softerware” that enables end users to manipulate the appearance and functionality of the charts and graphs they encounter according to their own preferences. We conducted empirical research to inform our collaborators, as well as other visualization system authors, with guidelines and considerations for building *softerware* systems. In our study, no two participants chose the same preference configuration and participants with the same diagnosed condition sometimes needed opposite design treatments. Practitioners, meanwhile, immediately raised ethical concerns about whether personalization would let designers off the hook for poor defaults. These findings revealed that the real barriers to personalization are not at the level of any individual chart but at the level of system infrastructure: without persistence, cross-system interoperability, and shared standards, the effort required of end users exceeds the value they receive.

Combined, these contributions provide empirical insights and practical advancements in the state of the art for tooling that bridges gaps in current accessibility practices in visualization and data science. Our work ultimately enables people with and without disabilities to better evaluate barriers in, analyze with, design for, develop, and personalize interactive data experiences. We demonstrate that tool-making is a productive intervention that both engages accessibility barriers and elucidates why those gaps exist in practitioner work.

Part III

Navigation: Making Data Structures Traversable

Chapter 5

Data Navigator: Low-level Tooling for Creating Navigable Data Structures

This chapter was adapted from my published paper:

F. Elavsky, L. Nadolskis, and D. Moritz, ‘Data Navigator: An Accessibility-Centered Data Navigation Toolkit’, *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–11, 2023.

5.1 Abstract

Making data visualizations accessible for people with disabilities remains a significant challenge in current practitioner efforts. Existing visualizations often lack an underlying navigable structure, fail to engage necessary input modalities, and rely heavily on visual-only rendering practices. These limitations exclude people with disabilities, especially users of assistive technologies. To address these challenges, we present Data Navigator: a system built on a dynamic graph structure, enabling developers to construct navigable lists, trees, graphs, and flows as well as spatial, diagrammatic, and geographic relations. Data Navigator supports a wide range of input modalities: screen reader, keyboard, speech, gesture detection, and even fabricated assistive devices. We present 3 case examples with Data Navigator, demonstrating we can provide accessible navigation structures on top of raster images, integrate with existing toolkits at scale, and rapidly develop novel prototypes. Data Navigator is a step towards making accessible data visualizations easier to design and implement.

5.2 Overview

While there is a growing interest in making data visualizations more accessible for people with disabilities, current toolkit and practitioner efforts have not risen to the challenge at scale. Major data visualization tools and ecosystems predominantly produce inaccessible artifacts for many users with disabilities. We believe this is largely a gap caused by a lack of underlying structure in most visualizations, failure to engage the input modalities used by people with disabilities, and over-reliance on visual-only rendering practices.

Users who are blind or low vision commonly use screen readers and users with motor and dexterity disabilities often do not use “pointer” (precise mouse and touch) based input technology when interacting with digital interfaces. Many users with motor and dexterity disabilities use discrete navigation controls, either sequentially using keyboard-like input, or directly using voice or text commands.

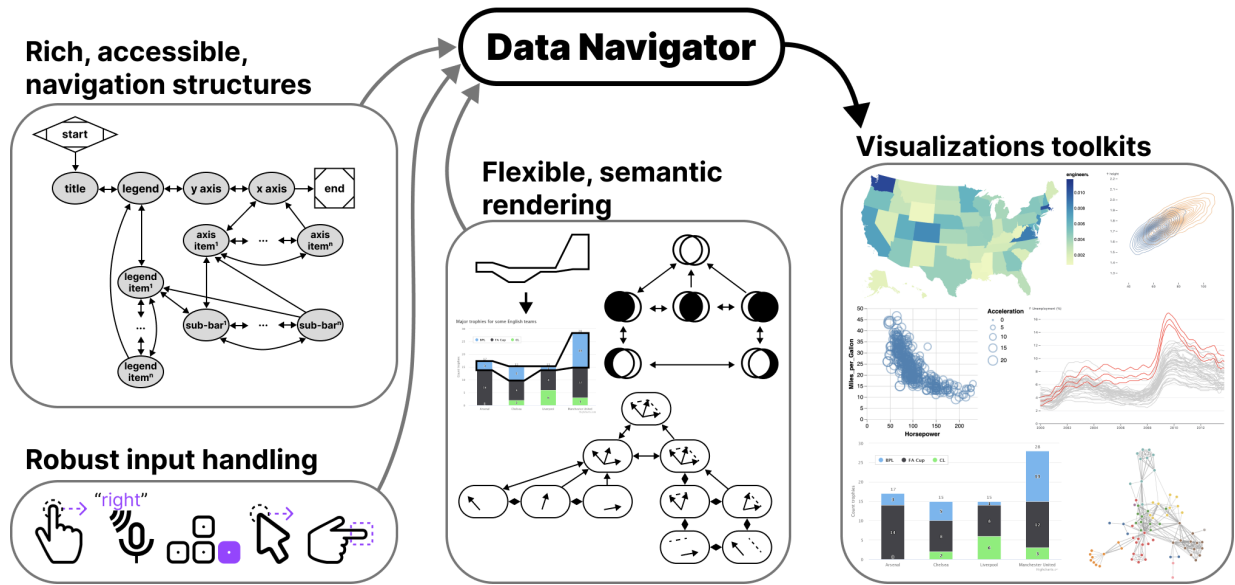


Figure 5.1: Data Navigator provides data visualization libraries and toolkits with accessible data navigation structures, robust input handling, and flexible semantic rendering capabilities.

Most interactive visualizations simply focus on pointer-based input: they can be clicked or tapped, hovered, and selected in order to perform analytical tasks. This excludes non-pointer input technologies. These devices require consideration for the navigation structure and underlying semantics of a visual interface.

However, building navigable spatial and relational interfaces is a difficult task with current resources.

Raster images, arguably the most common format for creating and disseminating data visualizations, currently cannot be made into navigable structures. These are only described using alt text, which limits their usefulness to screen reader users.

Unfortunately, more accessible rendering formats like SVG with ARIA (accessible rich internet applications) properties are more resource intensive than raster approaches, like WebGL-powered HTML canvas or pre-rendered PNG files. SVG puts a burden on low-bandwidth users and a ceiling on how many data points can be rendered in memory.

In addition, ARIA itself has 2 major limitations. First, when added to interface elements, ARIA only provides *screen reader* access, which means that developers must build a solution from scratch for other navigation input modalities. Second, ARIA’s linear navigation structure can be time-consuming for screen reader users if a visualization has many elements. This may impede how essential insights and relationships are understood [34, 60, 102, 108, 117, 141].

Some emerging approaches have sought to address this serial limitation of data navigation and provide richer experiences for screen reader users [34, 108, 117, 141]. However, these approaches rely on a tree-based navigation structure which is often not an appropriate choice for visualizations of relational, spatial, diagrammatic, or geographic data. Many visualization structures are currently unaddressed.

Zong et al. stress that in order to realize richer, more accessible data visualizations, the responsibility must be shared by “toolkit makers,” the practitioners who design, build, and maintain

visualization authoring technologies [141]. Our contribution is towards that aim, to make more accessible data experiences easier to design and implement within existing visualization work.

We present Data Navigator. Data Navigator is a toolkit built on a graph data structure, within which a broad array of common data structures can be expressed (including list, tree, graph, relational, spatial, diagrammatic, and geographic structures). Data Navigator also exposes an interface that supports interactions via screen reader, keyboard, gesture-based touch, motion gesture, voice, as well as fabricated and DIY input modalities. Data Navigator provides expressive structure and semantic rendering capabilities as well as the ability for developers to use their own, preferred method of rendering.

Data Navigator builds upon human-studies motivated work on accessible navigation [117, 141] towards a more generalizable resource for visualization practitioners. We contribute a high-level system design for our node-edge graph-based solution as well as an implementation of this system on the web, using JavaScript, HTML, and CSS. Through our case examples we also demonstrate that our generalized approach is suitable for replication of existing best practices from other systems, integration into existing visualization toolkit ecosystems, and development of novel prototypes for accessible navigation. We illustrate how Data Navigator’s use of generic edges, dynamic navigation rules, and loose coupling between navigation and visual encodings provides practitioners robust, expressive, control over their system designs.

5.3 Related Work

Our contribution is an attempt to bridge the gap between research and practice more effectively across broad ecosystems in order to enable deeper and more expressive accessible data navigation interfaces. Below we outline the prior research and standards that inform our project, a breakdown of existing visualization toolkit approaches to data navigation, and then accessible input device considerations.

5.3.1 Accessibility research and standards in visualization

Research and standards are both somewhat limited by a strong bias towards visual disabilities. In *Chartability*, 36 of the 50 criteria related to accessible visualization considerations involve visual disabilities [24, 27]. Marriott et al. also found that visual disability considerations are the primary focus of data visualization literature [78], leaving the barriers that many other demographics face unstudied.

However, despite the heavy focus on visual disabilities, the work that does exist in the visualization community is deeply valuable and serves as an important starting point for our technical contribution.

5.3.1.1 Accessible navigation design considerations

Zong et al.’s research, which was conducted as in-depth co-design work and validated in usability studies involving blind participants, presented a design space for accessible, rich screen reader navigation of data visualizations. They organized their design space into *structure*, *navigation*,

and *description* considerations and demonstrated example *structural*, *spatial*, and *direct* tree-based approaches [141].

Chart Reader also engaged these design space considerations in their co-design work on accessible data navigation structures [117]. We consider these design dimensions as the best starting point for our work, bridging the gap between research and toolkits.

There are additional research projects that have focused on accessible data navigation and interaction [34, 101, 103, 108]. These contributions explore a range of different interaction structures, including lists, trees, and tables of information as well as direct access methods such as voice interface commands and simple, pre-determined questions.

5.3.1.2 Accessible visualization: understanding users

A wide array of emerging research projects investigate screen reader users needs, barriers, and preferences, and offer guidelines, models, and considerations for creating accessible data visualizations [13, 27, 70, 102]. Jung et al. offer guidance to consider the order of information in textual descriptions and during navigation [60]. Kim et al. collected screen reader users' questions when interacting with data visualizations, which could open the door for more natural language data interaction [63].

5.3.1.3 Accessibility standards and guidelines

In the space of research, there has been a growing interest in developing guidelines for practitioners [22, 24] and even applying guidelines as a method of validation alongside human studies evaluations and co-design [27, 70, 71, 141]. Unfortunately, most accessibility standards and guidelines do not explicitly engage how to structure data navigation.

Despite this, existing accessibility standards bodies like the Web Content Accessibility Guidelines do stress the importance of accurate, functional semantics in order for screen reader users to know how to interact with elements [122]. For interactive visualizations this means that button-like or link-like behavior should expressly be made using elements that are semantically buttons and links. Our system should be capable of expressing meaningful semantics to users of assistive technologies.

5.3.2 Visualization toolkits and technical work

Unfortunately while many data visualization toolkits offer some degree of accessible navigation and interaction capabilities to developers, very few toolkits currently out there offer control over the important aspects of accessible data navigation design. Replicating existing research and strategies, remediating toolkit ecosystems, and building novel prototypes are all difficult or impossible to do due to the current lack of toolkit capabilities.

Existing data visualization toolkits have 3 major limitations that we wanted to address in the design of Data Navigator:

1. **Built on visual materials:** toolkits produce either raster or SVG-based visualizations, neither of which are focused towards designing navigable, semantic structures. As a consequence, many visualizations are simply entirely inaccessible.

2. **Lacking relational expressiveness:** When data navigation *is* provided, the navigation is based on either a tree or list structure (see Figure 5.2). The consequence of this limitation is that many other non-list and non-tree data relationships become difficult or impossible to represent without overly tedious navigation or inefficient architecture.
3. **Designed only for screen reader interaction:** When *accessible* data navigation is provided, it is generally only made possible through SVG with ARIA (Accessible Rich Internet Application) attributes. ARIA is primarily only leveraged by screen readers [123]. If a data element can be clicked and performs some form of function, only direct pointer (mouse and touch) and screen reader users are included. The consequence of this is that a wide array of other input devices, many used as assistive technologies by people with motor and dexterity disabilities, are excluded.

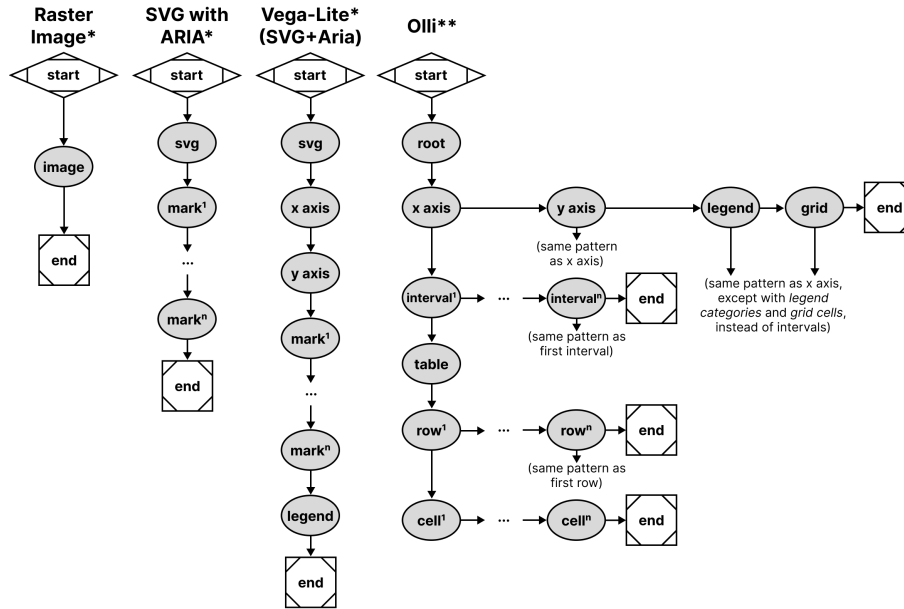


Figure 5.2: Existing accessibility trees and lists, shown using node-edge graph conventions. (*) Denotes only *screen reader* access. (**) Denotes *screen reader, keyboard-only, and pointer* access as well.

5.3.2.1 Rich, tree-based approaches

De-coupling rendered, visual structures from meaningful and effective navigation experiences can provide richer experiences for screen reader users [141]. Prior research and industry work, with the exception of the *Visa Chart Components* library [119], has relied heavily on a 1 to 1 relationship between structure (the encoded marks) and navigation. This emerging work is significant, because it paves the way for considering the design dimensions of accessible data interaction and navigation without dependence on a visually encoded space.

Olli's approach has been to build ready-to-go adaptors that automatically build multiple tree structures for a few ecosystems (*Vega*, *Vega-Lite*, and *Observable Plot*) and is entirely uncoupled

from a data visualization’s graphics. Their approach renders navigable tree structures *underneath* a visualization.

Other than *Olli*, *Highcharts* [50], *Visa Chart Components*, and *Progressive Accessibility Solutions*’ visualization toolkits [34, 108] also primarily provide tree and list navigation structures across all of their chart types. These toolkits render their structures *upon* the visualization’s graphic space. These tools also provide some degree of support for other assistive technologies and input modalities, although are limited exclusively to SVG rendering.

Unfortunately, these toolkits lack capabilities for dealing with graph, relational, spatial, diagrammatic, and geographic data structures.

5.3.2.2 Serial, list-based approaches

Toolkits like *Vega-Lite* [96] and *Observable Plot* only provide basic screen reader support through ARIA attributes when visualizations are rendered using SVG. These libraries do not currently provide additional access to other assistive technologies and input modalities.

Microsoft’s *PowerBI* largely uses a serial structure, although it has tree-like elements as well. *PowerBI* generally provides the same access to keyboard users as it does to screen readers, although not completely.

5.3.2.3 No navigation provided

Other visualization tools, like *ggplot2* or *Datawrapper*, *Tableau*, as well as both *Vega-Lite* and *Highcharts* (when rendering to canvas), produce raster images and have no navigable structure available. Raster, or pixel-based graphics have been an accessibility burden since the early days of graphical user interface development [8]. Practitioners who use these toolkits can only provide alternative text.

5.3.3 Considering assistive technologies and input devices

Modern data visualizations may contain functional capabilities such as the ability to hover, click, select, drag, or perform some analytical tasks over the elements of the visualization space [96]. Virtually all of these analytical capabilities are designed for use with a mouse.

Input device consideration can roughly be organized as either *pointer-based* (such as a mouse or direct touch) or *non-pointer based* (which may employ speech recognition or sequential, discrete navigation such as with a keyboard). Assistive pointer-based devices, such as a head-mounted touch stylus, can typically perform any actions that a mouse can and are therefore served by current interactive visualizations. However, assistive non-pointer devices, such as a tongue, foot, or breath-operated switch, are not.

By only providing pointer-based interactivity, modern interactive visualizations exclude users who leverage non-pointer based input, who are most commonly people with motor and dexterity disabilities. And unfortunately, there is a complete lack of engagement with these populations in the data visualization research community [78].

By comparison, the broader accessibility and HCI research communities have rich engagement with interaction and assistive technologies for users with motor and dexterity disabilities.

Most research either focuses broadly on physical peripheral devices or sensors [107], wearables [95], or DIY making and fabrication [54].

The DIY making space involves a broad spectrum of complex input devices and materials, such as fabricating with wood and sensors for children with disabilities [67], 3D printed materials for rehabilitation professionals [32], and even using produce-based input (such as bananas and cucumbers) for aging populations [89].

Broadly, both research and practical developments related to accessible, non-pointer input are much further ahead than data visualization research and practice. Our goal for Data Navigator is to provide a technical resource towards engaging this under-addressed space.

5.4 Data Navigator: System Design

We categorized our system design goals into design considerations for *Structure*, *Input*, and *Rendering*:

1. **Generic structure and navigation specification:** Human studies work has validated that lists, tables, trees, and even pseudo-treelike and direct structure types are all valuable to users in different contexts and with different considerations. Our system must be able to work with all of these as well as less frequently-used structures (spatial, relational, geographic, graph, and diagrammatic).
2. **Robust input handling:** Blind and low vision users may use combinations of different assistive technologies, such as magnifiers, voice interfaces, and screen readers. Users with motor impairments may rely on voice, gesture, eye-tracking, keyboard-interface peripherals (like sip-and-puffs or switches), or fabricated devices. Both the developer and user should therefore be able to leverage and customize a broad range of input types, including the above as well as fabricated, adaptive, and future input modalities.
3. **Flexible rendering and semantics:** Visuals may or may not be necessary to render to demonstrate Data Navigator’s structure. In addition, much of the latest research has shown that different screen reader users may prefer different orders of information and at different levels of verbosity. In addition, the context of tasks the user is performing as well as the nature of the data itself may influence the design of semantic descriptions and visual indications for elements. Data Navigator must provide a high degree of flexibility and control.

To help bridge the gaps between research and standards knowledge about best practices and building an effective toolkit for practitioners, we intend for Data Navigator to provide both *exploratory support* and *vocabulary correspondence* [72].

In particular, our ideal users are developers who specify data visualizations using code. To that aim, we intend to provide *exploratory support* through generic, dynamic, and flexible system design decisions. Our system is expressive and customizable, which encourages exploration of different options.

And we also want the API to include properties that have conceptual and *vocabulary correspondence* to our design considerations. Each design consideration (*Structure*, *Input*, and *Rendering*) are separately composable, modular subsystems of Data Navigator that can be used independently or in tandem with one another.

In this paper we present an implementation of our system using JavaScript, HTML, and CSS on the web. The demonstration of our system is best suited to the web due to the nature of existing, accessible building blocks (HTML), which resolve many of the semantic complexities and logic involved in enabling screen readers to programmatically navigate and announce meaningful information to users. In addition, many existing visualization toolkits target the web as an output platform and we believe that this is the best starting point for adoption and use of Data Navigator. However, this system design could be implemented as a toolkit in other environments with proper consideration for input device handling and screen reader semantics.

5.4.1 Structure

5.4.1.1 Beyond trees: towards an accessibility graph

The first major contribution in the design of Data Navigator is to use node-edge data as the substrate for our navigation system.

The most important argument in favor of using a graph-based approach is that a graph can construct virtually any other data structure type (see Figure 5.2), including list, table, tree, spatial, geographic, and diagrammatic structures. Graphs are generic, which enables them to represent structures both in current and future interface practices [31].

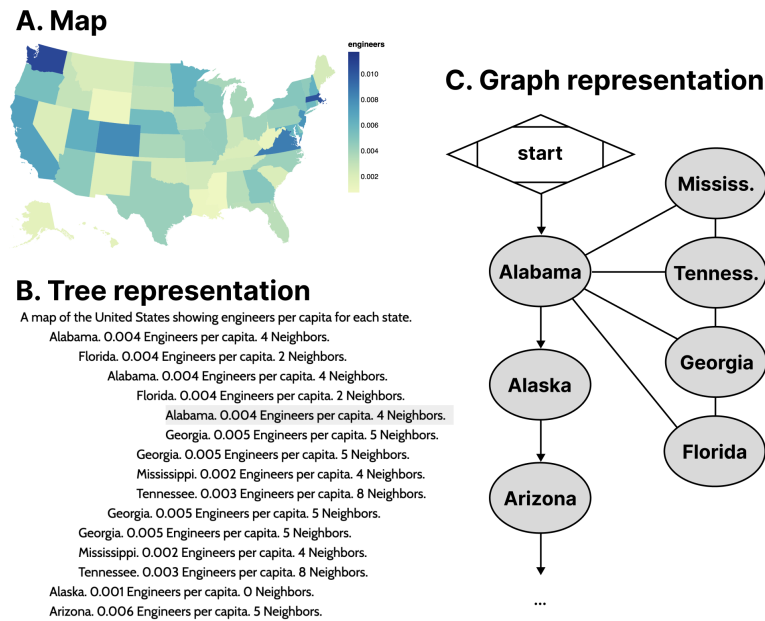


Figure 5.3: **A.** Map of engineers per capita of US states. **B.** Tree representation of the map data where states are listed alphabetically and also include links to neighboring states. The structure repeats itself if users navigate in a loop. **C.** Graph representation with the same navigation potential without redundant rendering.

To demonstrate our point, the most recent emerging work with advancements in accessible data navigation used node-edge diagrams to demonstrate their tree-like structures [117, 141]

similar to [Figure 5.2](#), [Figure 5.9](#), and [Figure 5.10](#). This is because trees are a form of node-edge graph, but with a root, siblings, parents, and children as sub-types of nodes that generally have rules for how they relate to one another.

Node-edge graph structures prioritize direct relationships. Examples of common direct relationships in visualization are boundaries on maps (see [Figure 5.3](#)), flows and cycles, data with multiple high level tree structures pointing to the same child datasets (such as *Olli* in [Figure 5.2](#)), or even just in diagrammatic, graph-based visualizations.

A graph structure allows for direct access between information elements that are not just part of the input data or 1:1 rendered elements, but may also have perceptual or human-attributed meaning. Examples of this might include semantic or task-based relationships, such as navigating to annotations or callouts, between visual-analytic features like trends, comparisons, or outliers. Spatial layouts such as intersections of sets or parallel vectors (see [Section 5.5.3](#)), or even relationships to information outside of a visualization and back into it (like in [Figure 5.7](#)) are enabled by a graph structure.

5.4.1.2 Graph structures are more computationally efficient

Data visualizations often portray information that becomes difficult to handle when using trees and lists. The distance users must travel between relational elements is significant in lists while redundancy when navigating relational elements in trees can be problematic.

As an example of this, often a data table or list of locations are used in conjunction to a map, such as listing all 50 states alphabetically along with relevant information. The list itself is expensive to navigate and may not provide any relationship information about which states border others, let alone ways to easily and directly access those states.

Part of the visual design justification of using a map instead of a table is for sighted individuals to understand how geospatial information may interact with a given variable. The spatial relationships matter. But when supplementing the list of states with sub-lists for each state's bordering states (see [Figure 5.3](#)), it produces redundancy in the rendered result. The rendered data contains circular connections between nodes but must render every reference, producing a computational resource creep and cluttered user experience that can be difficult to exit.

5.4.1.3 Specific edge instances and generic edges

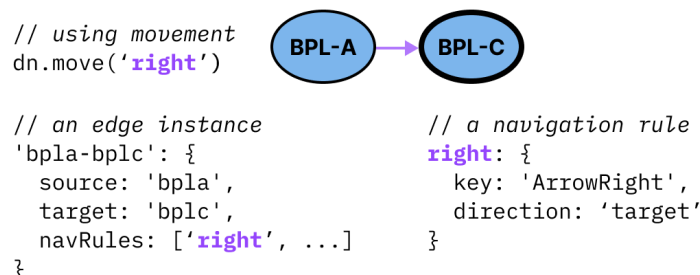


Figure 5.4: An example of how a single edge instance references a navigation rule and can even have multiple navigation rules. A navigation rule can be referenced by multiple edges.

In Data Navigator, nodes are *objects* that always contain a set of edges, where each edge contains a minimum of 4 pieces of information: a unique identifier, a source, a target, and navigation rules. These properties are only accessed when a navigation event occurs on a node with an edge that contains a reference to a rule for that navigation event. Navigation rules may be unique to an edge instance or shared among other edge instances.

The source and target properties of edges are either ids that reference node instances (see [Figure 5.4](#)) or *functions* (see [Figure 5.5](#)). Because some edges in a graph may be directed or not, non-directed graphs can use source and target properties to arbitrarily refer to either node attached to an edge.

Generic functions for source or target properties can link nodes to other nodes based on changing content, structure, or behavior that may be difficult or impossible to determine before a user navigates the structure.

Function calling also allows some edges to be *purely* generic. An example of a reasonable use case of a purely generic edge is in [Figure 5.5](#), where the source is a function which returns the present node and the target is whichever node the user was on previously. This single edge may then be part of every node’s set of edges, enabling users to have a simple *undo* navigation control without creating an *undo* edge unique to every source node.

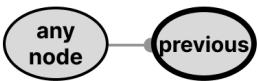
Using this pattern, it is possible to have fully navigable structures using only generic edges.

```

// using movement
dn.move('previous position')

// a generic edge
'any-return': {
  source: (_d, current, _p) => current,
  target: (_d, _c, previous) => previous,
  navRules: ['previous position', ...]
}

```



```

// a navigation rule
'previous position': {
  key: 'Period',
  direction: 'target'
}

```

Figure 5.5: A generic edge, such as “any-return” can be applied to any node. Function calls handle dynamically assigning the edge’s source and target nodes on-demand.

5.4.2 Input

5.4.2.1 Abstracted navigation facilitates agnostic input

Navigation rules in Data Navigator (see [Figure 5.4](#) and [Figure 5.5](#)) are created alongside the node-edge structure. Edges reference rules for navigation. However, these rules are generic and agnostic to the specifics of input modalities and can be invoked as methods by virtually any detected user input event (see [Figure 5.6](#)).

Navigation rules are objects with a unique name, ideally as a noun or verb in natural language that refers to a direction or location, a movement direction (a binary used to determine moving towards the source or target of an edge), and optionally any known user inputs that activate that navigation, such as a keyboard keypress event name.

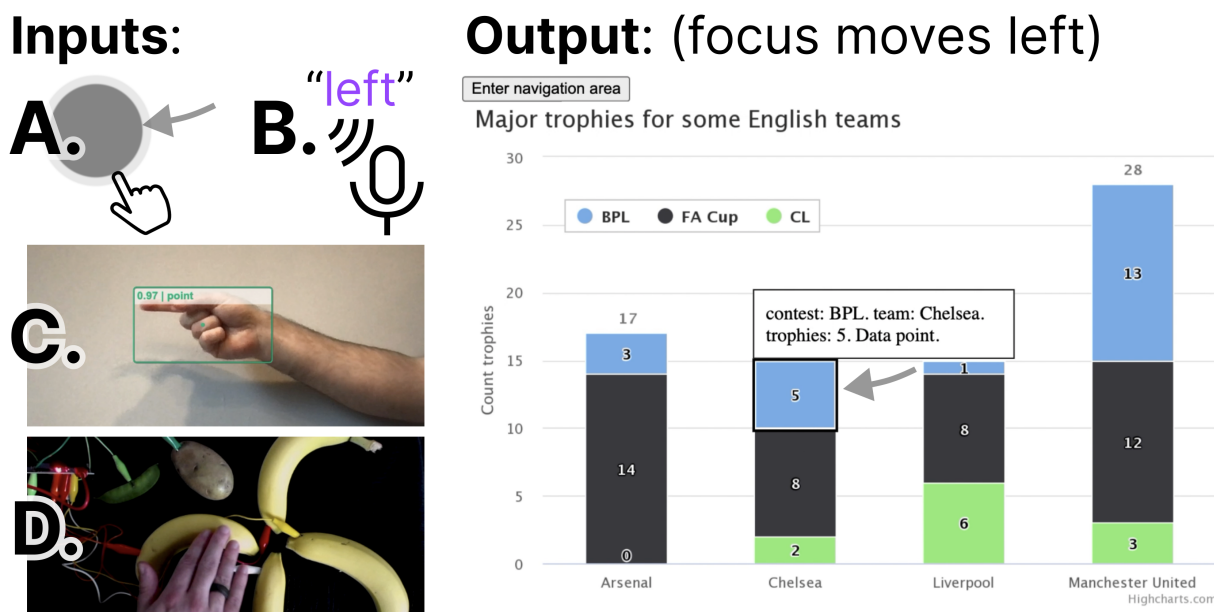


Figure 5.6: An example navigation rule to move “left” can be called as a method by an event from any input modality. Some examples include common modalities such as touch swiping (A) or speaking “left” (B). This also includes advanced or future modalities such as gesture recognition (C) or touch-activated, fabricated interfaces (D).

It is important for a system to abstract navigation events so that inputs can be uncoupled from the logic of Data Navigator. This allows higher level software or hardware logic to handle input validation while Data Navigator is just responsible for acting on validated input.

Later in our first case example (Section 5.5.1), we demonstrate an application that handles screen reader, keyboard, mouse and touch (pointer) swiping, hand gestures, typed text, and speech recognition input. Abstract navigation namespaces can be called by any of these input methods.

Additionally, since navigation rules are flexible, end users can also supply their own key-bind remapping preferences or input validation rules if developers provide them with an interface.

Because calling a navigation method is abstract, users can even supply events from their own input modalities as long they have access to either a text input interface or access to Data Navigator’s navigation methods. Our demonstration material (in Section 5.5.1) also includes handling for DIY fabricated interfaces, which are important in accessibility maker spaces. We chose a produce-based interface [89], since it was an easy and low cost proof of concept.

We believe that enabling agnostic input provides a rich space for future research projects. In addition, browser addons and assistive technologies could both leverage this flexible interface for end users.

5.4.2.2 Discrete, sequential input opens new avenues

The *keyboard interface* is considered foundational for many assistive devices, which leverage this technology for discrete, sequential, non-pointer navigation and interaction [124]. Desktop

5.4.3 Rendering

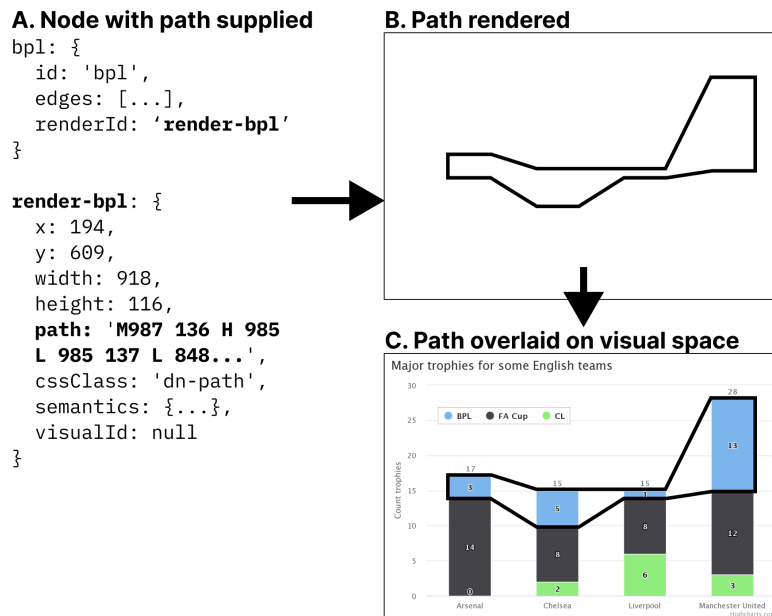


Figure 5.8: **A.** The data specified for a node with a reference to separate data that is used to render that node. **B.** The node will render as a path at the specified Cartesian coordinates. **C.** This rendered node may then be placed over a visual.

5.4.3.1 Flexible node semantics provide freedom

Nodes in Data Navigator are semantically flexible. This is because the marks in a data visualization may represent many things, that are either dependent on the data or the user interface materials.

Since our toolkit implementation is in JavaScript and HTML, our map example from [Figure 5.3](#) might use image semantics for states, alongside a description of the data relevant to that node. However since semantics are flexible in this way, Data Navigator could also be used to integrate into a larger ecosystem, with nodes rendered as hyperlinks to tables or other elements such as in [Figure 5.7](#).

The concept of using node-edge graphs can even extend to have “nodes” that are entirely different parts of a document or tool, as well as integrated into the explicit structure provided by Data Navigator. In some accessibility toolkits, nodes are geometries without functional semantics [96] or list items nested within lists [6]. But in Data Navigator, nodes can semantically be buttons, links, or any HTML element. Interactive data visualizations sometimes demand more flexible node semantics than geometries or lists.

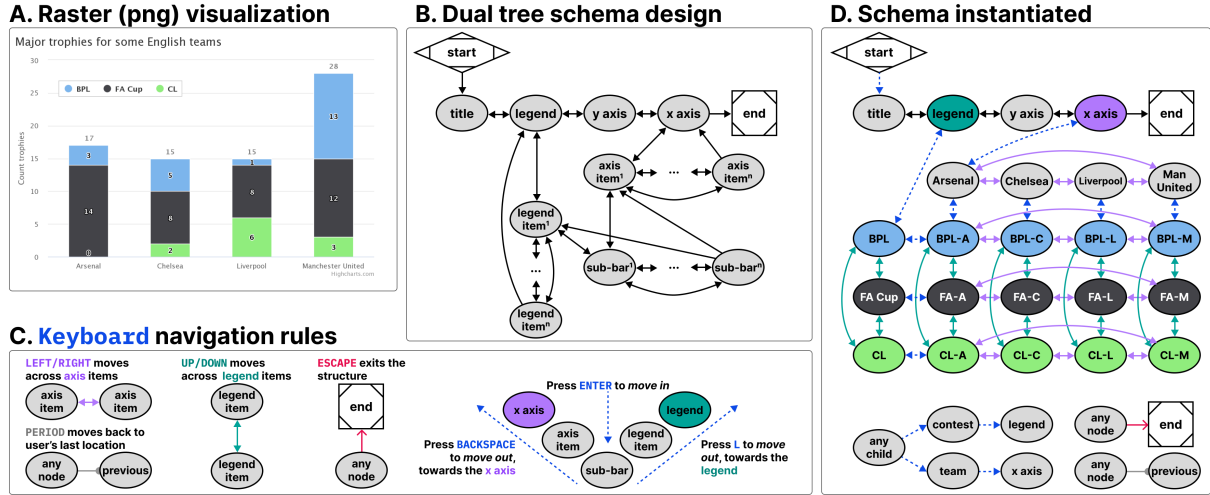


Figure 5.9: **A.** A raster (png) visualization of a stacked bar chart showing how 4 English teams performed across 3 major trophy contests. **B.** An example navigation schema that allows children nodes to have 2 parents (two tree structures intersecting), one for contests and one for teams. **C.** An example of Data Navigator’s navigation logic abstraction, which allows edge types to have programmatic sources, targets, and rules, such as a single rule that gives all nodes a edge to exit the visualization. **D.** An instantiation of the schema, showing all corresponding rendered nodes and their edge types according to the schema design and navigation rules.

5.4.3.2 Loose-coupling to visuals enables expressiveness

One of the most significant technical limitations of existing data visualization toolkits with regards to accessibility is that they rely on visual substrate, or visual materials, in order to produce data visualizations. In the case of static, raster images such as png files or WebGL and canvas elements on the web, there are no interface properties at all exposed to screen readers for programmatic exploration and interaction.

If raster images are used, they generally cannot be changed after rendering. However, according to web accessibility standards, elements must have a visual indicator provided when focused [121].

Since Data Navigator navigates using focus, an indicator must be rendered alongside the node semantics. But *what* is focused visually and *where* it is depends on different design needs.

In *Visa Chart Components*, chart elements can be *selected*, so the focus indication is visible over the existing elements in the chart space. The design choice to have interactive visual elements located within a chart or graph is also common in other toolkits that provide accessible focus indication, such as *Highcharts*, *PowerBI*, and *SAS Graphics Accelerator*.

However, some visualization toolkits create accessible structures entirely uncoupled from visual space [6], so focus indication is provided beneath or beside the chart, not over it.

Due to the different ways that accessibility might be provided, Data Navigator enables developers to have complete control over the rendering of which focus elements they want, in what styling, and where. This can accommodate both un-coupled and visually-coupled approaches to focusing and more.

Data Navigator’s focus is *uncoupled* by default and may even be used independent of any existing graphics at all. Rendering information may be passed to Data Navigator for it to render (like in [Figure 5.8](#)) or developers can provide their own rendered elements and simply use Data Navigator to move between them.

Because of Data Navigator’s approach to rendering focusable elements, designers and developers can provide fully customized annotations, graphics, text, or marks that may not be part of the original visual space or elements. One example of this might be adding an outlined path to a collective cross-stack group of bars in a stacked bar chart (see [Figure 5.8](#)).

Loose-coupling in this way provides robust flexibility to designers and developers to handle navigation paths and stories through a data visualization, even in bespoke or hand-crafted ways.

5.4.3.3 On-demand node rendering is efficient

Practitioners care about performance and so do users. Practitioner toolkits often focus on lazy-loading techniques where accessibility elements are rendered on-demand rather than all in-memory up front [6, 23, 141].

Data Navigator’s nodes are rendered *on-demand* by default. Data Navigator only renders the node that is about to be focused by the user and after it is focused, the previously focused node is deleted from memory. This technique has advantages in cases where datasets are large or users have lower computational bandwidth available. However, there are cases where practitioners may want to render all of Data Navigator’s structure in memory, such as server-side rendering or equivalent. Pre-rendering may be optionally enabled.

5.5 Case Examples with Data Navigator

We built example prototypes using our JavaScript implementation of Data Navigator, available open source at our [GitHub repository](#).

Our first two prototype case examples represent some of the most powerful parts of Data Navigator as a system while reproducing known and effective data navigation patterns from existing industry and research projects. We provide a final case example as a co-design session that demonstrates how Data Navigator may be used to rapidly build new designs.

5.5.1 Augmenting a Static, Raster Visualization

The first case example (shown in [Figure 5.9](#)) builds on an online JavaScript visualization library, *Highcharts*. *Highcharts* already provides relatively robust data navigation handling out of the box for screen reader, keyboard, and even voice recognition interface technologies, such as *Dragon Naturally Speaking*. However, these capabilities are only provided when the chart is rendered using SVG. Developers have several other rendering options available, including WebGL, which is significantly more efficient [51]. We wanted to demonstrate that Data Navigator can provide a navigable data structure even if the underlying visualization is a raster image.

For our case example, we exported a png file using the built in menu of a sample stacked bar chart retrieved from their online demos [49]. We selected a stacked bar chart because it allows

us to demonstrate how two tree structures may interact and share the same children nodes.

We recorded the data and hand-created all of the geometries and their spatial coordinates using *Figma*, by tracing lines over the raster image’s geometries (see samples of the data and traced geometries in [Figure 5.8](#)). While this method was efficient for building an initial prototype, [Section 5.5.2](#) engages deterministic methods for extracting and producing the nodes, edges, and descriptions required by Data Navigator automatically and at scale.

The visualization we selected represents 4 English football teams, *Arsenal*, *Chelsea*, *Liverpool*, and *Manchester United* and how many trophies they won across 3 contests, *BPL*, *FA Cup*, and *CL*.

We chose a schema design that arranged the *contests* to be navigable across one dimension of movement (*up* and *down*) while the *teams* are navigable across a perpendicular dimension of movement (*left* and *right*). This 2-axis style of navigation is used by *Highcharts* (when rendering as SVG) and *Visa Chart Components*. We also chose these directions because it is coincidental that their visual affordance is closely coupled with the navigation design (the x axis is ordered *left* to *right* and since the bars are stacked, *up* and *down* can move within the stack). These directions can also be applied to the axis categories and legend categories as well, moving *left* and *right* across the entire *team*’s stacks or *up* and *down* across the entire *contest*’s groupings.

Using a keyboard, a user might enter this schema and navigate to the legend, where they could press *Enter* to then focus the legend’s first child, pictured in [Figure 5.8](#). Pressing *up* or *down* navigates in a circular fashion among the *contest* groupings. Pressing *Enter* again then focuses the first child element of that *contest*, all of which are in the *Arsenal* group, since it is the first group along the x axis. A user can then navigate *up*, *down*, *left*, and *right* among children. Pressing *L* Key moves the user back up towards the contest while pressing *Backspace* moves the user up towards the x axis. The x axis and *team* groupings represent the second tree which intersects the first (the *contests*).

Our first case example includes handling for additional input modalities beyond screen readers and keyboards, including a hand gesture recognition model, swipe-based touch navigation, and text input (which can be controlled using voice recognition software).

5.5.1.1 Discussion

Our first case example demonstrates several of the most important capabilities of Data Navigator, namely that practitioners can add accessible navigation to previously inaccessible, static, raster image formats and that a wide variety of input modalities are supported easily.

Widely-used toolkits like *Vega-Lite*, *Highcharts*, and *D3* [\[7\]](#) allow practitioners to choose SVG and canvas-based rendering methods. Data Navigator’s affordances help overcome the lack of semantic structure in canvas-based rendering, allowing developers to take advantage of its processing and memory efficiency.

Notably in addition to these capabilities, the visual focus highlighting added was entirely bespoke (as in [Figure 5.8](#)) and the navigation paths through the visual were based on our design intentions, not an extracted view or underlying architecture such as render order. This demonstrates that our system provides a significant degree of freedom and control for designers and developers.

As a final discussion point, the resulting visualization contains no automatically detectable

accessibility conformance failures according to the W3C’s Web Accessibility Initiative’s accessibility evaluation tool, *WAVE* [126]. It is important for any technology developed to also meet minimum requirements for accessibility [24, 70, 71, 141], even when following best-practices and research.

5.5.2 Building Data Navigation for a Toolkit Ecosystem

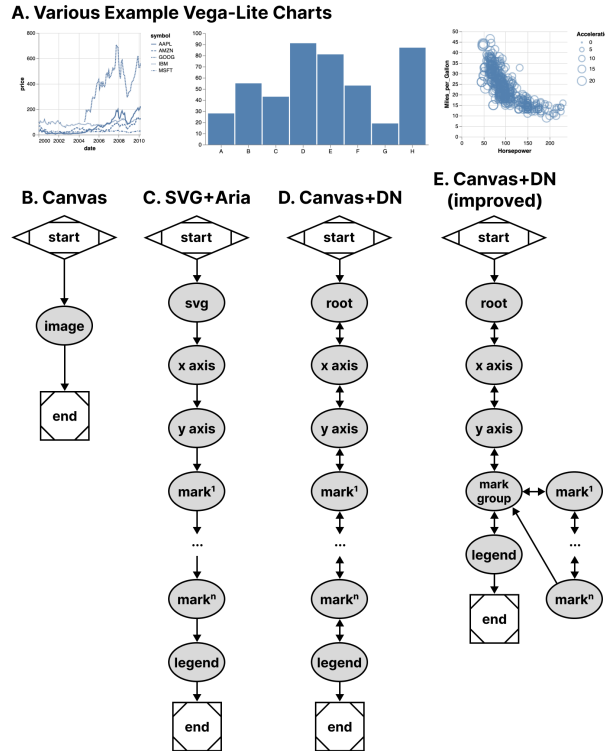


Figure 5.10: **A.** Various charts from *Vega-Lite* share the same general structures with each other when rendered using canvas (**B**) or SVG (**C**). **D.** With Data Navigator, we replicated the existing SVG navigation pattern (**C**) but used a canvas-based rendering for the visualization. **E.** We also improved the navigation scheme to nest marks within a mark group to allow users to skip them, if needed.

Our second case example, shown in Figure 5.10, builds on *Vega-Lite*. As shown in Figure 5.2, *Vega-Lite* offers basic screen reader navigation but provides no navigation at all when rendered using canvas.

While it might be a tedious design choice to allow every mark in a visualization to be serially accessible to screen reader users, we nevertheless set out to build a generic ingestion function that would take a *Vega-Lite View* object and deterministically recreate their existing SVG navigation structure in Data Navigator. This way users would have the same experience between SVG rendered charts and all current and future rendering options that *Vega-Lite* offers to developers.

Notably, *Vega-Lite* does not explicitly manipulate the navigation order at all when rendering with SVG. ARIA is simply provided to allow screen reader users to access each mark in the

visualization in the order the mark appears in the DOM (which is the order it was rendered). The legend appears after the marks in our schema for this reason because *Vega-Lite* renders the legend after marks. This choice of ordering is for visual reasons: z-axis placement is currently based on render order in SVG and *Vega-Lite* wants their legend visually on top of the rendered marks.

In addition to mimicking their existing SVG navigation strategy, we also created a way to nest all of the marks within a group so that users can skip past them and drill in on-demand, which is a valuable pattern when dealing with situations where providing a mark-level fidelity of information may not be relevant to a user's needs by default [106, 141].

In order to deterministically supply Data Navigator with accurate information about any given *Vega-Lite* visualization, we built 3 functions: one that takes a *Vega-Lite View* as input and extracts meaningful nodes, one that produces edges based on those nodes, and one to describe our nodes in a meaningful way for screen reader users. These generic functions technically work on all existing *Vega-Lite* charts, however some are more useful out of the box than others due to the type of marks involved.

5.5.2.1 Discussion

This case example demonstrates that ecosystem-level remediation and customization is not only possible for toolkit builders but Data Navigator offers robust potential. Data Navigator's structure, input, and rendering capabilities are all flexible and can be adjusted to suit the needs of a specific toolkit's design and intended use.

Many visualization libraries may not even provide screen reader accessible SVG using ARIA-based approaches but do have a consistent underlying architectural pattern. Some libraries have a consistent method for converting data into visual formats, readable text labels, and interaction logic. Strong contenders would be visualization libraries popular in online, web-based data science notebooks like *ggplot2* in R or *matplotlib* for Python, which typically only render rasterized pngs or semanticless SVG.

Toolkits with consistent underlying architecture would allow toolkit developers, not just developers who *use* toolkits, to remediate and customize their navigation accessibility using a generic approach.

Enabling accessibility at the toolkit level allows all downstream use of that tool to have better defaults, options, and resources available for building more accessible outcomes for end users.

Many libraries and toolkits provide users with a level of functional defaults and abstract conciseness so that users don't have to worry about low-level geometric considerations [96].

Data Navigator allows toolkit developers to also provide their users with abstractions and defaults for accessibility that make sense for their ecosystem.

Despite our schema recreating a screen reader experience based on SVG (and improving it), Data Navigator's additional features also apply: users are able to leverage a much wider array of input modalities.

Vega-Lite provides many ways to make marks clickable and even perform complex actions using mouse-based input. While Data Navigator does not engage accessible brush and drag-based inputs, it does provide keyboard-only access by default, which can be used to make events

previously only accessible to mouse clicking available to many other technologies. This is an improvement over *Vega-Lite*'s SVG + ARIA rendering option.

When measuring performance across test datasets containing 406 and 20,300 data points in a scatter plot, Data Navigator increases initialization time by ~ 0.45 to ~ 1.5 ms respectively. Our extraction functions specific to *Vega-Lite* increase initialization between ~ 4.8 and ~ 8.5 ms respectively. Given that our benchmark testing for *Vega-Lite*'s SVG rendering initialized in $\sim 1,800$ ms for 20,300 data points and canvas in ~ 700 ms, we do not anticipate that Data Navigator will have a negative impact on performance in most visualization contexts.

5.5.3 Co-designing Novel Data Navigation Prototypes

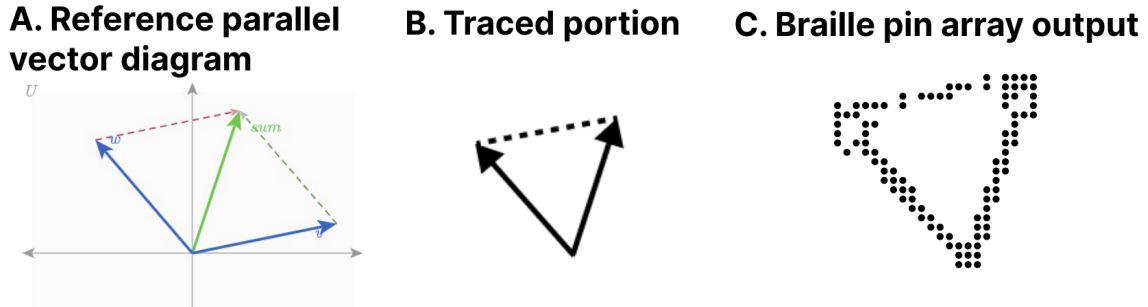


Figure 5.11: Our material preparation process involved taking a reference (A), tracing it (B), and rendering it on a tactile display (C).

Recent projects in accessible data navigation have involved extensive co-design work with people with disabilities, ranging on the magnitude of months with as many as 10 co-designers at a time [70, 71, 117, 141].

However many visualization experiences may be authored in smaller scales, with fewer designers, and less time such as the development of a prototype or demonstration of an emerging idea. In practical or industry contexts, co-design sessions (and design sessions in general) may be much shorter. The goal of these co-design sessions is simply to create an artifact with the artifact's intended users.

Since our paper is contribution towards practical outcomes, we simulated a light co-design session with the aim of producing low-fidelity prototypes of novel data interaction patterns.

5.5.3.1 Co-design Session Methods and Setup

Authors Frank Elavsky (sighted) and Lucas Nadolskis (blind) set out with the goal of developing screen-reader friendly prototypes that can explore geometric and mathematical models produced by the math diagramming tool *Penrose* [138].

Nadolskis is a neuroscience engineer who is a native screen reader user and uses both mathematical concepts as well as data-related tasks in his research. Elavsky proposed a series of possible math-based visualization types produced by *Penrose* to build prototypes for, and Nadolskis selected *set* and *vector* diagrams as the two worth exploring first. The justification for this

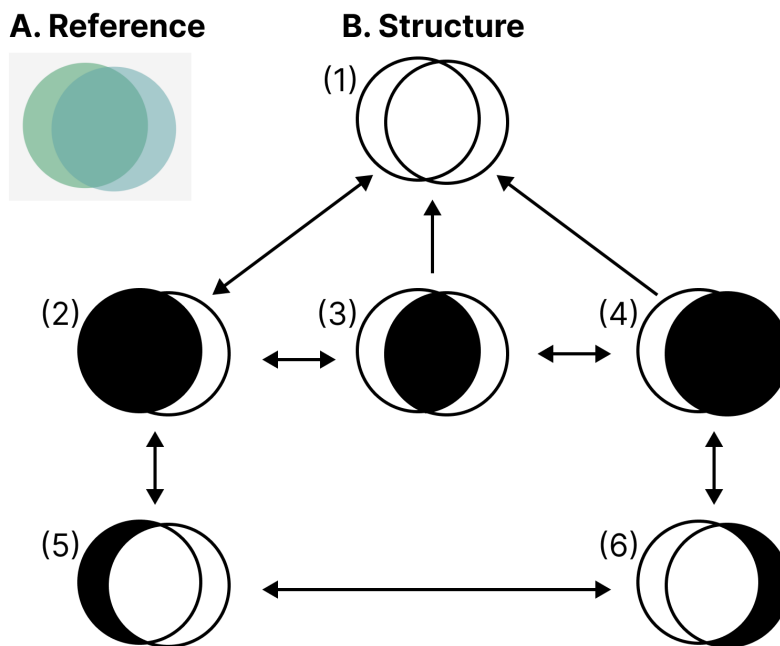


Figure 5.12: **A.** A reference image from *Penrose* of a set diagram containing two sets intersecting. **B.** A diagram of our proposed structure, with three levels of information.

selection is that understanding these two concepts is important for work in data science, programming, and more advanced math concepts.

In particular we grounded the context of our contribution in a hypothetical classroom setting, where a screen reader user who is a student will have access to the equations in both raw text and *MathJax*. We want to provide an experience that does not replace the existing resources screen reader users have to learn in classrooms but rather supplement.

At our disposal for our co-design session was a *Dot Pad* [19], which is a refreshable tactile braille display. Our *Dot Pad* enabled Elavsky to produce something visual and then translate it into the display for Nadolskis. Similar to de Greef et al. [16], we used a tactile interface as an intermediary to help us get a shared sense of the meaningful spatial features of our figures.

Elavsky started with a reference diagram and then traced a wide variety of every possible node that might be worth navigating to in the diagram (see Figure 5.11).

We selected which nodes were most important in each diagram, how to navigate between them, and how we wanted to render their visuals and semantics.

The selection of our problem space, scope of solutions, context of contribution, general discussion, and preparation of materials took approximately 12 hours of work over 2 weeks. The exploration of our prototype design space for our 2 prototypes took 1 hour. Building the prototypes took 2 hours.

5.5.3.2 Creating a Navigable Set Diagram

Our first prototype was a set diagram (see Figure 5.12). For our structure, we decided that it has 3 important semantic levels: the high level, the inclusion level, and the exclusion level. The

inclusion level is first and the siblings are all sets or subsets that include other sets. The exclusion level is beneath and contains sets or subsets that are exclusive to the sets they belong to, which are accessed by drilling down from a set.

Our schema design starts with a user encountering the root level (1) and may optionally drill in to the first child of the next level (2) using the *Enter* key. The user may navigate siblings at this level using *right* and *left* directions, but this level is not circular (like in Figure 5.9) to maintain the spatial relationships. The user may drill in on either set again to view the non-intersecting portion of that set. Any node can drill up, towards the root, using *Escape* or *Backspace*.

5.5.3.3 Creating a Navigable Parallel Vectors Diagram

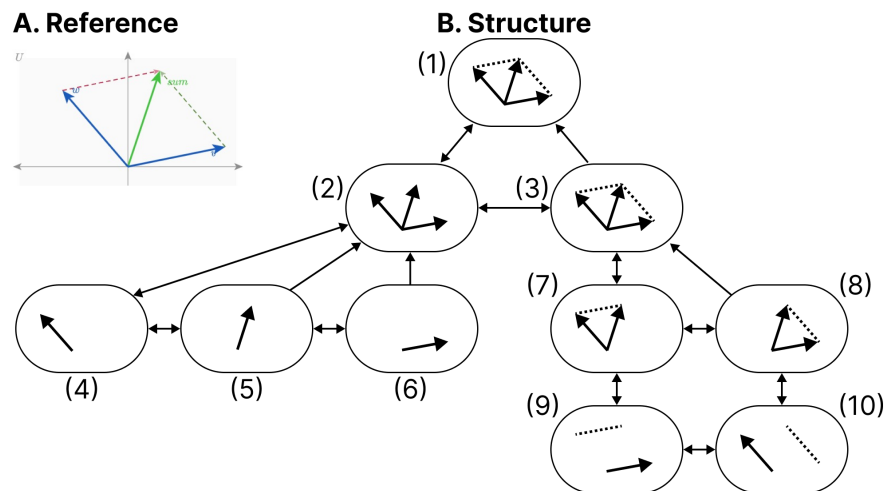


Figure 5.13: **A.** A reference image from *Penrose* of a parallel vectors diagram. **B.** A diagram of our proposed structure, with two main sub-categories of information: understanding the vectors and their parallels.

Our second prototype was a parallel vectors diagram (see Figure 5.13). For the structure of this diagram we created a first level group that contains each vector and vector sum. The sibling to this grouping is another group which organizes sub-equations related to calculating each parallel vector. The sub equations each contain children that pair the sub equation with the vector it is parallel to.

Similar to Figure 5.12, this figure maintains spatial relationships along the x dimension, does not have circular navigation, and allows drilling in and out.

5.5.3.4 Discussion

After our co-design sessions, our visual materials and navigation structures were used in the creation of functional prototypes. We additionally hand-crafted the descriptions and semantics for each node.

Accessibility work often takes a long time, from co-design to building to validation. But we believe that a well-articulated and useful design space, with tools that provide expressiveness

and control over the dimensions of that design space, can improve how this work is done. The above case example demonstrates how builders who are thinking about data navigation design can rapidly scaffold prototypes for use in Data Navigator.

In particular, Data Navigator’s design as a system gave our co-design sessions *vocabulary correspondence*. Data Navigator’s language helped us focus on the *nodes*, *edges*, and *navigation rules* for our *structure* while we also explicitly discussed the *rendering* details of *coordinates*, *shapes*, *styling*, and *semantics* for each node. The vocabulary of our design space directly corresponded with code details required to create a functional prototype.

We note that this co-design work is not intended to contribute a *validated* set of designs. Rather, our contribution with this case example is to demonstrate that within the larger ecosystem of a research venture, Data Navigator is an improvement over designing and building navigable structures from scratch.

5.6 Limitations and Future Work

Data Navigator is a technical contribution, a system designed for appropriation [20] and adaptation [133] in different applied contexts. It is, as Louridas writes, a *technical material*: a technology that enables new and useful capabilities [69]. While beyond the scope of the current paper, a critical next step for future work is to conduct separate studies with both practitioners and end users to evaluate Data Navigator’s affordances.

Unlike toolkits that provide an end-to-end development pipeline for accessible visualization, Data Navigator serves as a low-level building block or material (like concrete). As such, one potential limitation of the framework is that it can be used to build both curbs (which are inaccessible) as well as ramps and *curb-cuts* (which may be more broadly accessible).

Even when building more accessible curb-cuts, we stress the importance of actively involving people with disabilities in the design and validation of new ideas, in line with prior work [70, 71, 88, 141]. For example, while our first two case examples replicate co-designed and validated existing work, our third case example’s co-designed prototypes would need to be validated with relevant stakeholders before wider implementation. Our system does not *guarantee* any sort of accessibility on its own.

The diverse array of modalities supported by Data Navigator opens an immediate line of future work in engaging people with a correspondingly diverse set of disabilities. While recent explorations into accessible data visualization have been inspiring, this trend has primarily focused on the experiences of people with visual disabilities [24, 64, 78]. More research should be conducted with other populations, particularly people who leverage assistive technologies beyond screen readers, to understand how interactive data visualizations can be better designed to serve them.

Finally, there are significant opportunities to improve the efficiency of our approach, including developing deterministic and non-deterministic methods to generate node-edge data and navigation rules from a visualization. Ma’ayan et al. stress in particular that reducing tedious complexity can contribute to the success of a well-designed toolkit [72]. Future work should identify areas where graphical interface tools or higher-level specifications can improve the experience of working with Data Navigator.

5.7 Conclusion

Practitioners at large continue to produce inaccessible interactive data visualizations, excluding people with disabilities. We believe that the burden of remediation first starts with the developers who build and maintain the toolkits that practitioners use.

However, the challenges faced by toolkit builders are significant. Most toolkits lack an underlying, navigable structure, support for broad input modalities used by people with disabilities, and meaningful, semantic rendering.

To engage these limitations we present Data Navigator, a technical contribution that builds on existing work towards a more generalizable accessibility-centered toolkit for creating data navigation interfaces. Data Navigator is designed for use by practitioners who both build and use existing toolkits and want a tool to make their data visualizations and interfaces more accessible.

We contribute a high-level system design for our node-edge graph-based approach that can be used to build data structures that are navigable by a wide array of assistive technologies and input modalities. Data Navigator is generic and can scaffold list, tree, graph, relational, spatial, diagrammatic, and geographic types of data structures common to data visualization.

Our system is designed to encourage both remediation of existing inaccessible systems and visualization formats as well as help scaffold the design of novel, future projects. We look forward to further research that explores the possibilities enabled by Data Navigator.

Part VI

Conclusion

Chapter 9

Discussion & Future Work

9.1 What is a “tool?” A reflection on the social and material identity of tools

In the introduction of this dissertation, I use the example of a hammer: a hammer can destroy and it can construct. So is the *use* of a technology what constitutes it? Do we understand the hammer as the *thing we swing, to destroy and to build*? Should we?

This thesis engages domains of tools and tool-making for accessibility: evaluation, navigation, interaction, and personalization. But these categories for work do not fully characterize the upstream conditions that our software systems and data interfaces inherit.

In my work specifically on accessibility, a larger social reality becomes apparent that shapes the question, “what is a tool?” far more than how an individual might use one, or the domains of work that our tool-making inhabits. My research journey has navigated multiple social and political thresholds, from changes to law in the European Union, to the enactment of Title II as part of the update to the Americans with Disabilities Act. These laws have motivated a significant interest in accessibility research, solutions, guidelines, and technologies. In the midst of this, we have seen the rise of overlays and generative AI solutionism [42] and subsequent lawsuits and grass-roots resistance.

For my work, this is mostly good news. Legal change produces motivation, and even with predatory technology attempting to address real problems, pushback is widespread and active. But this paints a picture of the reality that my work inherits: many tools cannot even be used, or cease to be used, if there is not a social, political, and material set of conditions in place motivating those tools, providing resources for their construction, regulating their use, and examining the outcomes of what they accomplish. Tools and technologies are often a response to social, cultural, political, and legal realities that we first negotiate.

I recently spoke on this at a keynote in Australia, on how a hammer isn’t *just* a tool and that the idea that “the only thing that matters is how a tool is used” limits how we really understand tools. Instead, I spoke about how a standard, household hammer requires iron and wood. That alone leads to a whole universe of different questions. Western Australia’s conservation efforts were disrupted when a significant amount of natural iron was discovered in a wildlife preserve. So laws were passed and now iron is mined there. That iron is largely exported. And Australia then, whether with Australian iron or not, mostly imports their small tools. Iron is sent out, and through a complex network of trade (likely indirectly related to the iron), hammers are brought in. A “hammer,” to even exist at all, relies on multiple levels of human governance, international relations, and complex infrastructures of trade.

And while my metaphor is largely motivated to encourage younger practitioners to consider the “iron mines” in the technologies they use, such as modern generative AI, it is also an area that is not adequately explored and addressed in terms of accessibility research.

Research on accessibility is dependent on funding, which is often dependent on political

priorities and action. Depending on the current social and political state of the world at large, accessibility research itself may never gain the opportunities required in order to innovate and produce new tools at all. And as the US’s 2025 federal cuts to research demonstrated, millions of dollars devoted to accessibility research can be lost to political agendas. It is for this reason that engagement with policy recommendation and guidance is essential. Personal political activity and involvement is also essential. Researchers who genuinely believe in accessibility as a human right or as a dignity that all people deserve should work with policymakers to ensure that there are material and structural resources in place for this work to continue. We cannot naively believe that technology, divorced from the realm of social and political forces, is capable of solving accessibility barriers [111]. Without enforcement and threat of litigation, very little accessibility work has been accomplished in the past by technology companies alone.

Not featured in these chapters (as they were merely stapled in research papers from previous publications) is the policy and outreach work involved in seeing that work like *Chartability* and *Data Navigator* are used in real contexts, including by organizations that govern and influence the lives of many people. Immediate incentives to produce novelty may not be enough to sustain the larger socio-cultural and political ecosystems that our work participates in and is downstream from. We must also get involved.

9.3 Who is responsible for repair?

Lastly, I want to revisit one of my opening points, where I argue that the *tool-makers* are first responsible for repair. This is true. However, the most pressing issue I have faced in recent years is mostly unmentioned across these research projects: tool-makers might be responsible, but this is because they are the only ones who have the *power* to make things accessible. Does this always need to be the case? Can we imagine an artifact’s authority over the user’s ability to access being designed towards self-subversion [35] or de-centralized agency [11, 62, 81], instead? What might that look like, concretely?

In *Softerware*, we begin to engage this larger problem in terms of an idealized state where a user can repair or re-design their own experiences. But to me, this self-repair is like laying down train tracks for yourself as you move a locomotive, but then lifting up your own tracks behind you as you go. You’re the only one helping yourself. This is not ideal, for you or others.

What we need are broad, lasting, infrastructural changes. On the web, this problem becomes quite difficult to solve. A personal computer or device? Again, someone can auto-design their interfaces into a better state. But back when I started *Chartability*, the WebAim Million’s report showed more than 95% of the top one million website home pages contain at least one critical accessibility error. And now, more than 6 years later, that proportion is unchanged [127].

Some had imagined that generative AI would solve the massive infrastructural repair problems we face. But unfortunately, the latest WebAim Million report shows that since 2020, ARIA usage has increased and correlates to more errors, while use of tabindex on a page has increased nearly 300% and also correlates to more errors on a page. If anything, during the age of generative AI, we have seen existing bad patterns worsen in prevalence and complexity.

I firmly believe that a tools-based approach is not enough on its own. Tool-making cannot be the *only* intervention on inaccessibility. Tools and tool-making, as our thesis argues, have a

powerful role to play. But we simply can't tool our way out of failed infrastructure and inadequate policy when someone else *owns* the tools and tool-making. Visiting a website is like going into someone else's home: arranged according to their effort, tastes, and so on. If you can't access their home, you essentially need to request that they let you in personally. Website repair always falls to the owner and maintainer of a website, and they largely don't take any meaningful action.

Sidewalks outside of homes are a good parallel to this problem. Sidewalk accessibility is a massive infrastructural problem [92], and yet localities treat sidewalk maintenance in different ways: some, like where I presently live in the south hills of Pittsburgh, put the onus on the homeowner whose house and property the sidewalk touches. In other places, sidewalks are considered a public path, similar to a roadway, and are maintained through public tax and resource management. To no surprise, privately-maintained, public-access sidewalks are worse for people in pretty much every way than publicly-maintained ones [134]. This is because private homeowners don't care about sidewalk maintenance unless the city manages to fine them or they get sued.

And the web is a collection of private spaces that you visit privately. There is no truly shared, universally democratic, public space on the web. Centralization is partly to blame: sharing space while scaling leads to consolidation.

So my future work will continue to wrestle with the same tensions of scale, repair, and anti-consolidation of power, motivated by the same WebAim Million report. But now I look to questions of *democratic* and *radical* access to accessibility repair. The barriers I hope to tackle in the future are political and infrastructural. Perhaps tool-making will participate in this work, but it seems clear now from my work that the upstream technical problems and socio-political conditions that tools inherit, will likely not be addressed by tools alone.

What does "democratic" and "radical" infrastructure work look like? It will probably be an extension of *Software*, to some degree. I imagine future research that explores public-first spaces, ones where access is socially negotiated and repair belongs to all of us. Is this an autonomous space, like an autonomous zone [5] separate from the web? Above it, looking down into it, like shared annotation tools but capable of sharing the manipulation of websites [91]? A space with ambient co-repair, modeled after projects that bring people together [100] or that allow community "fixing" of misinformation [56]? Perhaps feminist thought on the ethics of care can help us [45, 74]? Or maybe it will be something else entirely; I'm not yet sure. But what made the web fantastic years ago is long gone; most of it has been hedged into corporate spaces that are controlled, maintained, and repaired by corporate power. And these entities are notoriously bad at repair. What I imagine in the future involves reclaiming a sense that the web is *ours*, belongs to *us*, and that ultimately *we* are responsible for making it accessible.

References

- [1] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pages 111–117. IEEE, November 2005. doi: 10.1109/infvis.2005.1532136. URL <https://doi.org/10.1109/infvis.2005.1532136>. 3
- [2] Michel Beaudouin-Lafon, Susanne Bødker, and Wendy E. Mackay. Generative theories of interaction. *ACM Transactions on Computer-Human Interaction*, 28(6):1–54, December 2021. ISSN 1557-7325. doi: 10.1145/3468505. URL <https://doi.org/10.1145/3468505>. 2.1.2
- [3] Richard A. Becker and William S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, May 1987. ISSN 1537-2723. doi: 10.1080/00401706.1987.10488204. URL <https://doi.org/10.1080/00401706.1987.10488204>. 2.2.1
- [4] Dan Bennett, Alan Dix, Parisa Eslambolchilar, Feng Feng, Tom Froese, Vassilis Kostakos, Sebastien Lericque, and Niels van Berkel. Emergent interaction: Complexity, dynamics, and enaction in hci. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI ’21, pages 1–7. ACM, May 2021. doi: 10.1145/3411763.3441321. URL <https://doi.org/10.1145/3411763.3441321>. 2.1.2
- [5] Hakim Bey. *TAZ: The temporary autonomous zone, ontological anarchy, poetic terrorism*. Autonomedia, 2003. 9.3
- [6] Matt Blanco, Jonathan Zong, and Arvind Satyanarayan. Olli: An Extensible Visualization Library for Screen Reader Accessibility. In *IEEE VIS Posters*, 2022. URL <https://vis.csail.mit.edu/pubs/olli>. Accessed: 2026-03-30, <https://vis.csail.mit.edu/pubs/olli>. 5.4.3.1, 5.4.3.2, 5.4.3.3
- [7] M. Bostock, V. Ogievetsky, and J. Heer. D³ data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, December 2011. ISSN 1077-2626. doi: 10.1109/tvcg.2011.185. URL <https://doi.org/10.1109/tvcg.2011.185>. 1.2, 5.5.1.1
- [8] L.H. Boyd, W.L. Boyd, and G.C. Vanderheiden. The graphical user interface: Crisis, danger, and opportunity. *Journal of Visual Impairment & Blindness*, 84(10):496–502, December 1990. doi: 10.1177/0145482x9008401002. URL <https://doi.org/10.1177/0145482x9008401002>. 5.3.2.3
- [9] S. Brewster. Visualization tools for blind people using multiple modalities. *Disability and Rehabilitation*, 24(11-12):613–621, January 2002. ISSN 1464-

5165. doi: 10.1080/09638280110111388. URL <https://doi.org/10.1080/09638280110111388>. 2.2.3
- [10] Jennifer L. Burke, Matthew S. Prewett, Ashley A. Gray, Liuquin Yang, Frederick R. B. Stilson, Michael D. Covert, Linda R. Elliot, and Elizabeth Redden. Comparing the effects of visual-auditory and visual-tactile feedback on user performance. In *Proceedings of the 8th international conference on Multimodal interfaces*, ICMI06, pages 108–117. ACM, November 2006. doi: 10.1145/1180995.1181017. URL <https://doi.org/10.1145/1180995.1181017>. 1.1
 - [11] Katherine Casey. Radical decentralization: Does community-driven development work? *Annual Review of Economics*, 10(1):139–163, August 2018. ISSN 1941-1391. doi: 10.1146/annurev-economics-080217-053339. URL <http://dx.doi.org/10.1146/annurev-economics-080217-053339>. 9.3
 - [12] Alex Chaparro and Maria Chaparro. Applications of Color in Design for Color-Deficient Users. *Ergonomics in Design*, 25(1):23–30, 1 2017. doi: 10.1177/1064804616635382. Accessed: 2021-09-06. 2.2.3
 - [13] Pramod Chundury, Biswaksen Patnaik, Yasmin Reyazuddin, Christine Tang, Jonathan Lazar, and Niklas Elmqvist. Towards understanding sensory substitution for accessible visualization: An interview study. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1084–1094, January 2022. doi: 10.1109/tvcg.2021.3114829. URL <https://doi.org/10.1109/tvcg.2021.3114829>. 2.2.3, 5.3.1.2
 - [14] Pramod Chundury, Urja Thakkar, Yasmin Reyazuddin, J. Bern Jordan, Niklas Elmqvist, and Jonathan Lazar. Understanding the visualization and analytics needs of blind and low-vision professionals. In *The 26th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS ’24, pages 1–5. ACM, October 2024. doi: 10.1145/3663548.3688496. URL <https://doi.org/10.1145/3663548.3688496>. 2.1.1
 - [15] Clare Cullen and Oussama Metatla. Co-designing inclusive multisensory story mapping with children with mixed visual abilities. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children*, IDC ’19, pages 361–373. ACM, June 2019. doi: 10.1145/3311927.3323146. URL <https://doi.org/10.1145/3311927.3323146>. 2.2.3
 - [16] Lilian de Greef, Dominik Moritz, and Cynthia Bennett. Interdependent variables: Remotely designing tactile graphics for an accessible workflow. In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility*, pages 1–6. ACM, October 2021. doi: 10.1145/3441852.3476468. URL <https://doi.org/10.1145/3441852.3476468>. 5.5.3.1
 - [17] Deque. Automated Testing Identifies 57 Percent of Digital Accessibility Issues. Technical report, Deque, 3 2021. Accessed: 2021-09-06. 3
 - [18] Alan Dix. Designing for appropriation. In *Electronic Workshops in Computing*. BCS Learning & Development, September 2007. doi: 10.14236/ewic/hci2007.53. URL <https://doi.org/10.14236/ewic/hci2007.53>. 2.1.2

- [19] Dot Pad inc. Dot pad - the first tactile graphics display for the visually impaired. <https://pad.dotincorp.com/>, 2020. [Accessed 01-04-2025]. 5.5.3.1
- [20] Paul Dourish. The appropriation of interactive technologies: Some lessons from placeless documents. *Computer Supported Cooperative Work (CSCW)*, 12(4):465–490, December 2003. doi: 10.1023/a:1026149119426. URL <https://doi.org/10.1023/a:1026149119426>. 5.6
- [21] Sebastian Draxler, Gunnar Stevens, Martin Stein, Alexander Boden, and David Randall. Supporting the social context of technology appropriation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2835–2844. ACM, May 2012. doi: 10.1145/2207676.2208687. URL <https://doi.org/10.1145/2207676.2208687>. 2.1.2
- [22] Eloi Durant, Mathieu Rouard, Eric W. Ganko, Cedric Muller, Alan M. Cleary, Andrew D. Farmer, Matthieu Conte, and Francois Sabot. Ten simple rules for developing visualization tools in genomics. *PLOS Computational Biology*, 18(11):e1010622, November 2022. doi: 10.1371/journal.pcbi.1010622. URL <https://doi.org/10.1371/journal.pcbi.1010622>. 2.2.2.2, 5.3.1.3
- [23] Frank Elavsky. Method and system for accessible data visualization on a web platform. *Defensive Publication Series*, 4220, April 2021. 5.4.3.3
- [24] Frank Elavsky, Cynthia Bennett, and Dominik Moritz. How accessible is my visualization? evaluating visualization accessibility with chartability. *Computer Graphics Forum*, 41(3):57–70, June 2022. doi: 10.1111/cgf.14522. URL <https://doi.org/10.1111/cgf.14522>. 2.2.2.1, 2.2.2.2, 2.2.3, 5.3.1, 5.3.1.3, 5.5.1.1, 5.6
- [25] Frank Elavsky, Lucas Nadolskis, and Dominik Moritz. Data navigator: An accessibility-centered data navigation toolkit. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–11, 2023. ISSN 2160-9306. doi: 10.1109/tvcg.2023.3327393. URL <https://doi.org/10.1109/tvcg.2023.3327393>. 2.1.2
- [26] Will Epperson, Vaishnavi Gorantla, Dominik Moritz, and Adam Perer. Dead or alive: Continuous data profiling for interactive data science. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–11, 2023. ISSN 2160-9306. doi: 10.1109/tvcg.2023.3327367. URL <https://doi.org/10.1109/tvcg.2023.3327367>. 2.2.1
- [27] Danyang Fan, Alexa Fay Siu, Hrishikesh Rao, Gene Sung-Ho Kim, Xavier Vazquez, Lucy Greco, Sile O’Modhrain, and Sean Follmer. The accessibility of data visualizations on the web for screen reader users: Practices and experiences during covid-19. *ACM Transactions on Accessible Computing*, 16(1):1–29, March 2023. doi: 10.1145/3557899. URL <https://doi.org/10.1145/3557899>. 1.1, 2.2.2.1, 2.2.2.2, 2.2.3, 5.3.1, 5.3.1.2, 5.3.1.3
- [28] Stephen Few. Visual Business Intelligence - Data Visualization and the Blind. <https://www.perceptualedge.com/blog/?p=1756>, 2013. [Accessed 05-04-2025]. 1.1
- [29] John H. Flowers, Dion C. Buhman, and Kimberly D. Turnage. Cross-Modal Equivalence of Visual and Auditory Scatterplots for Exploring Bivariate Data Samples. *Human Fac-*

tors, 39(3):341–351, 9 1997. doi: 10.1518/001872097778827151. Accessed: 2021-09-03. 2.2.3, 3

- [30] Steven L. Franconeri, Lace M. Padilla, Priti Shah, Jeffrey M. Zacks, and Jessica Hullman. The science of visual data communication: What works. *Psychological Science in the Public Interest*, 22(3):110–161, December 2021. ISSN 1539-6053. doi: 10.1177/15291006211051956. URL <https://doi.org/10.1177/15291006211051956>. 1.2
- [31] Emden R. Gansner and Stephen C. North. An open graph visualization system and its applications to software engineering. *Software: Practice and Experience*, 30(11):1203–1233, September 2000. doi: 10.1002/1097-024x(200009)30:11<1203::aid-spe338>3.0.co;2-n. URL [https://doi.org/10.1002/1097-024x\(200009\)30:11<1203::aid-spe338>3.0.co;2-n](https://doi.org/10.1002/1097-024x(200009)30:11<1203::aid-spe338>3.0.co;2-n). 5.4.1.1
- [32] Stéphanie Giraud and Christophe Jouffrais. Empowering low-vision rehabilitation professionals with “do-it-yourself” methods. In *Lecture Notes in Computer Science*, pages 61–68. Springer International Publishing, 2016. ISBN 9783319412665. doi: 10.1007/978-3-319-41267-2_9. URL https://doi.org/10.1007/978-3-319-41267-2_9. 5.3.3
- [33] Michele E. Gloede and Melissa K. Gregg. The fidelity of visual and auditory memory. *Psychonomic Bulletin & Review*, 26(4):1325–1332, August 2019. ISSN 1531-5320. doi: 10.3758/s13423-019-01597-7. URL <https://doi.org/10.3758/s13423-019-01597-7>. 1.1
- [34] A. Jonathan R. Godfrey, Paul Murrell, and Volker Sorge. An accessible interaction model for data visualisation in statistics. In *Lecture Notes in Computer Science*, pages 590–597. Springer International Publishing, 2018. ISBN 9783319942766. doi: 10.1007/978-3-319-94277-3_92. URL https://doi.org/10.1007/978-3-319-94277-3_92. 2.2.3, 5.2, 5.3.1.1, 5.3.2.1
- [35] David Graeber and Charlie Rose. Never cower to authority, 2006. URL <https://www.youtube.com/watch?v=qt5op6wft-8>. Accessed [2026]. 9.3
- [36] David Gray Widder, Sarah West, and Meredith Whittaker. Open (for business): Big tech, concentrated power, and the political economy of open ai. *SSRN Electronic Journal*, 2023. ISSN 1556-5068. doi: 10.2139/ssrn.4543807. URL <https://doi.org/10.2139/ssrn.4543807>. 1.2
- [37] Catherine Grevet and Eric Gilbert. Piggyback prototyping. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI ’15, pages 4047–4056. ACM, April 2015. doi: 10.1145/2702123.2702395. URL <https://doi.org/10.1145/2702123.2702395>. 2.1.2
- [38] Nathan Hahn, Joseph Chang, Ji Eun Kim, and Aniket Kittur. The knowledge accelerator. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI’16, pages 2258–2270. ACM, May 2016. doi: 10.1145/2858036.2858364. URL <https://doi.org/10.1145/2858036.2858364>. 2.1.1
- [39] Jen Hale. Extensive digitization of tactile map collection.

<https://www.perkins.org/extensive-digitization-of-tactile-map-collection/>, 2016. Perkins Archives Blog, Perkins School for the Blind. July 2016. Accessed: 2022-02-23. 2.2.3

- [40] Foad Hamidi, Patrick Mbullo, Deurence Onyango, Michaela Hynie, Susan McGrath, and Melanie Baljko. Participatory design of diy digital assistive technology in western kenya. In *Proceedings of the Second African Conference for Human Computer Interaction: Thriving Communities*, AfriCHI '18, pages 1–11. ACM, December 2018. doi: 10.1145/3283458.3283478. URL <https://doi.org/10.1145/3283458.3283478>. 2.1.1
- [41] A. Hamraie. Making access critical: Disability, race, and gender in environmental design. <https://belonging.berkeley.edu/aimi-hamraie-making-access-critical-disability-race-and-gender-environment> 2019. [Online]. 2.2.3, 3
- [42] Parker Hartman and Tim Gorichanaz. Evaluating ai-powered website accessibility overlays. In *Proceedings of the 27th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '25, page 1–4. ACM, October 2025. doi: 10.1145/3663547.3759761. URL <http://dx.doi.org/10.1145/3663547.3759761>. 9.1
- [43] Jeffrey Heer and Maneesh Agrawala. Design considerations for collaborative visual analytics. In *2007 IEEE Symposium on Visual Analytics Science and Technology*, pages 171–178. IEEE, October 2007. doi: 10.1109/vast.2007.4389011. URL <https://doi.org/10.1109/vast.2007.4389011>. 2.2.1
- [44] Jeffrey Heer and Dominik Moritz. Mosaic: An architecture for scalable & interoperable data views. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–11, 2023. ISSN 1077-2626, 1941-0506, 2160-9306. doi: 10.1109/tvcg.2023.3327189. URL <https://doi.org/10.1109/tvcg.2023.3327189>. 1.2, 2.2.1, 3
- [45] Ana O Henriques, Anna R. L. Carter, Beatriz Severes, Reem Talhouk, Angelika Strohmayer, Ana Cristina Pires, Colin M. Gray, Kyle Montague, and Hugo Nicolau. A feminist care ethics toolkit for community-based design: Bridging theory and practice. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, page 1–26. ACM, April 2025. doi: 10.1145/3706598.3713950. URL <http://dx.doi.org/10.1145/3706598.3713950>. 9.3
- [46] Jaylin Herskovitz, Yi Fei Cheng, Anhong Guo, Alanson P. Sample, and Michael Nebeling. Xspace: An augmented reality toolkit for enabling spatially-aware distributed collaboration. *Proceedings of the ACM on Human-Computer Interaction*, 6(ISS):277–302, November 2022. ISSN 2573-0142. doi: 10.1145/3567721. URL <https://doi.org/10.1145/3567721>. 2.1.2
- [47] Jaylin Herskovitz, Andi Xu, Rahaf Alharbi, and Anhong Guo. Hacking, switching, combining: Understanding and supporting diy assistive technology design by blind people. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, pages 1–17. ACM, April 2023. doi: 10.1145/3544548.3581249. URL <https://doi.org/10.1145/3544548.3581249>. 2.1.2

- [48] Jaylin Herskovitz, Andi Xu, Rahaf Alharbi, and Anhong Guo. Programally: Creating custom visual access programs via multi-modal end-user programming. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST '24, pages 1–15. ACM, October 2024. doi: 10.1145/3654777.3676391. URL <https://doi.org/10.1145/3654777.3676391>. 2.1.2
- [49] Highsoft. Stacked column demo. <https://www.highcharts.com/demo/column-stacked>, 2018. Accessed: 2022-12-31. 5.5.1
- [50] Highsoft. Highcharts accessibility module: information and demos. <https://highcharts.com/docs/accessibility/accessibility-module>, 2018. Accessed: 2021-09-06. 5.3.2.1
- [51] Highsoft. Highcharts: Render millions of chart points with the boost module. <https://www.highcharts.com/blog/tutorials/highcharts-high-performance-boost-module/>, 2020. Accessed: 2022-11-20. 5.5.1
- [52] Jonathan Hook, Sanne Verbaan, Abigail Durrant, Patrick Olivier, and Peter Wright. A study of the challenges related to diy assistive technology in the context of children with disabilities. In *Proceedings of the 2014 conference on Designing interactive systems*, DIS '14, pages 597–606. ACM, June 2014. doi: 10.1145/2598510.2598530. URL <https://doi.org/10.1145/2598510.2598530>. 2.1.1
- [53] Stacy Hsueh, Beatrice Vincenzi, Akshata Murdeshwar, and Marianela Ciolfi Felice. Crippling data visualizations: Crip technoscience as a critical lens for designing digital access. In *The 25th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '23, page 1–16. ACM, October 2023. doi: 10.1145/3597638.3608427. URL <http://dx.doi.org/10.1145/3597638.3608427>. 1.1, 2.2.3, 3
- [54] Amy Hurst and Shaun Kane. Making "making" accessible. In *Proceedings of the 12th International Conference on Interaction Design and Children*, IDC '13, pages 635–638, New York, NY, USA, 6 2013. Association for Computing Machinery. doi: 10.1145/2485760.2485883. Accessed: 2021-09-03. 2.1.1, 5.3.3
- [55] Amy Hurst and Jasmine Tobias. Empowering individuals with do-it-yourself assistive technology. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, ASSETS '11, pages 11–18. ACM, October 2011. doi: 10.1145/2049536.2049541. URL <https://doi.org/10.1145/2049536.2049541>. 2.1.1
- [56] Farnaz Jahanbakhsh, Amy X. Zhang, Karrie Karahalios, and David R. Karger. Our browser extension lets readers change the headlines on news articles, and you won't believe what they did! *Proceedings of the ACM on Human-Computer Interaction*, 6 (CSCW2):1–33, November 2022. ISSN 2573-0142. doi: 10.1145/3555643. URL <http://dx.doi.org/10.1145/3555643>. 9.3
- [57] Chutian Jiang, Wentao Lei, Emily Kuang, Teng Han, and Mingming Fan. Understanding strategies and challenges of conducting daily data analysis (dda) among blind and low-vision people. In *The 25th International ACM SIGACCESS Conference on Computers*

- and Accessibility, ASSETS '23, pages 1–15. ACM, October 2023. doi: 10.1145/3597638.3608423. URL <https://doi.org/10.1145/3597638.3608423>. 2.1.1
- [58] Shuli Jones, Isabella Pedraza Pineros, Daniel Hajas, Jonathan Zong, and Arvind Satyanarayan. “customization is key”: Reconfigurable textual tokens for accessible data visualizations. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, pages 1–14. ACM, May 2024. doi: 10.1145/3613904.3641970. URL <https://doi.org/10.1145/3613904.3641970>. 3
- [59] Shakila Cherise S Joyner, Amalia Riegelhuth, Kathleen Garrity, Yea-Seul Kim, and Nam Wook Kim. Visualization accessibility in the wild: Challenges faced by visualization designers. In *CHI Conference on Human Factors in Computing Systems*, CHI '22, pages 1–19. ACM, April 2022. doi: 10.1145/3491102.3517630. URL <https://doi.org/10.1145/3491102.3517630>. 2.2.3, 3
- [60] Crescentia Jung, Shubham Mehta, Atharva Kulkarni, Yuhang Zhao, and Yea-Seul Kim. Communicating visualizations without visuals: Investigation of visualization alternative text for people with visual impairments. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1095–1105, January 2022. doi: 10.1109/tvcg.2021.3114846. URL <https://doi.org/10.1109/tvcg.2021.3114846>. 2.2.3, 5.2, 5.3.1.2
- [61] Hyeonsu B. Kang, Xin Qian, Tom Hope, Dafna Shahaf, Joel Chan, and Aniket Kittur. Augmenting scientific creativity with an analogical search engine. *ACM Transactions on Computer-Human Interaction*, 29(6):1–36, December 2022. ISSN 1557-7325. doi: 10.1145/3530013. URL <https://doi.org/10.1145/3530013>. 2.1.1
- [62] Os Keyes, Josephine Hoy, and Margaret Drouhard. Human-computer insurrection: Notes on an anarchist hci. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–13. ACM, May 2019. doi: 10.1145/3290605.3300569. URL <http://dx.doi.org/10.1145/3290605.3300569>. 9.3
- [63] Jiho Kim, Arjun Srinivasan, Nam Wook Kim, and Yea-Seul Kim. Exploring chart question answering for blind and low vision users. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–15. ACM, April 2023. doi: 10.1145/3544548.3581532. URL <https://doi.org/10.1145/3544548.3581532>. 2.2.3, 5.3.1.2
- [64] N. W. Kim, S. C. Joyner, A. Riegelhuth, and Y. Kim. Accessible visualization: Design space, opportunities, and challenges. *Computer Graphics Forum*, 40(3):173–188, June 2021. doi: 10.1111/cgf.14298. URL <https://doi.org/10.1111/cgf.14298>. 2.2.3, 5.6
- [65] Kitware, Inc. *The Visualization Toolkit user’s guide*, January 2003. URL <http://www.kitware.com/publications/item/view/1269>. 2.1.2
- [66] Bongshin Lee, Eun Kyoung Choe, Petra Isenberg, Kim Marriott, and John Stasko. Reaching broader audiences with data visualization. *IEEE Computer Graphics and Applications*, 40(2):82–90, 2020. doi: 10.1109/MCG.2020.2968244. 2.2.3
- [67] Chien-Yu Lin and Yu-Ming Chang. Increase in physical activities in kindergarten children with cerebral palsy by employing makey–makey-based task systems. *Research in Devel-*

- opmental Disabilities*, 35(9):1963–1969, September 2014. doi: 10.1016/j.ridd.2014.04.028. URL <https://doi.org/10.1016/j.ridd.2014.04.028>. 5.3.3
- [68] Zhicheng Liu and Jeffrey Heer. The effects of interactive latency on exploratory visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2122–2131, December 2014. ISSN 1077-2626. doi: 10.1109/tvcg.2014.2346452. URL <https://doi.org/10.1109/tvcg.2014.2346452>. 2.2.1, 3
- [69] Panagiotis Louridas. Design as bricolage: anthropology meets design thinking. *Design Studies*, 20(6):517–535, November 1999. doi: 10.1016/s0142-694x(98)00044-1. URL [https://doi.org/10.1016/s0142-694x\(98\)00044-1](https://doi.org/10.1016/s0142-694x(98)00044-1). 5.6
- [70] Alan Lundgard and Arvind Satyanarayan. Accessible visualization via natural language descriptions: A four-level model of semantic content. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1073–1083, January 2022. doi: 10.1109/tvcg.2021.3114770. URL <https://doi.org/10.1109/tvcg.2021.3114770>. 2.2.2.2, 2.2.3, 5.3.1.2, 5.3.1.3, 5.5.1.1, 5.5.3, 5.6
- [71] Alan Lundgard, Crystal Lee, and Arvind Satyanarayan. Sociotechnical considerations for accessible visualization design. In *2019 IEEE Visualization Conference (VIS)*, pages 16–20. IEEE, October 2019. doi: 10.1109/visual.2019.8933762. URL <https://doi.org/10.1109/visual.2019.8933762>. 2.2.2.2, 5.3.1.3, 5.5.1.1, 5.5.3, 5.6
- [72] Dor Ma’ayan, Wode Ni, Katherine Ye, Chinmay Kulkarni, and Joshua Sunshine. How domain experts create conceptual diagrams and implications for tool design. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI ’20, pages 1–14, New York, NY, USA, April 2020. ACM. ISBN 9781450367080. doi: 10.1145/3313831.3376253. URL <https://doi.org/10.1145/3313831.3376253>. 5.4, 5.6
- [73] Kelly Mack, Emma McDonnell, Dhruv Jain, Lucy Lu Wang, Jon E. Froehlich, and Leah Findlater. What do we mean by “accessibility research”? In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI ’21, pages 1–18, New York, NY, USA, May 2021. ACM. ISBN 978-1-4503-8096-6. doi: 10.1145/3411764.3445412. URL <https://doi.org/10.1145/3411764.3445412>. Accessed: 2022-02-22. 2.2.2.1, 2.2.3
- [74] Els Maeckelberghe. Feminist ethic of care: A third alternative approach. *Health Care Analysis*, 12(4):317–327, December 2004. ISSN 1573-3394. doi: 10.1007/s10728-004-6639-6. URL <http://dx.doi.org/10.1007/s10728-004-6639-6>. 9.3
- [75] R G Maling and D C Clarkson. Electronic controls for the tetraplegic (possum) (patient operated selector mechanisms—p.o. s.m.). *Spinal Cord*, 1(3):161–174, November 1963. ISSN 1476-5624. doi: 10.1038/sc.1963.15. URL <http://dx.doi.org/10.1038/sc.1963.15>. 3
- [76] Douglass L. Mansur, Merra M. Blattner, and Kenneth I. Joy. Sound graphs: A numerical data analysis method for the blind. *Journal of Medical Systems*, 9(3):163–174, June 1985. ISSN 1573-689X. doi: 10.1007/bf00996201. URL <https://doi.org/10.1007/bf00996201>.

- [77] Deborah Marks. Models of disability. *Disability and Rehabilitation*, 19(3):85–91, January 1997. ISSN 1464-5165. doi: 10.3109/09638289709166831. URL <https://doi.org/10.3109/09638289709166831>. 1.1
- [78] Kim Marriott, Bongshin Lee, Matthew Butler, Ed Cutrell, Kirsten Ellis, Cagatay Goncu, Marti Hearst, Kathleen McCoy, and Danielle Albers Szafr. Inclusive data visualization for people with disabilities. *Interactions*, 28(3):47–51, May 2021. doi: 10.1145/3457875. URL <https://doi.org/10.1145/3457875>. 2.2.1, 2.2.2.1, 2.2.3, 5.3.1, 5.3.3, 5.6
- [79] Rubén Alcaraz Martínez, Mireia Ribera Turró, and Toni Granollers Saltiveri. Methodology for heuristic evaluation of the accessibility of statistical charts for people with low vision and color vision deficiency, February 2021. URL <https://doi.org/10.21203/rs.3.rs-156959/v1>. 2.2.3
- [80] David K. McGookin and Stephen A. Brewster. Soundbar: Exploiting multiple views in multimodal graph browsing. In *Proceedings of the 4th Nordic conference on Human-computer interaction: Changing roles*, NordiCHI '06, pages 145–154, New York, NY, USA, 10 2006. Association for Computing Machinery. doi: 10.1145/1182475.1182491. Accessed: 2021-09-03. 2.2.3
- [81] Giles Mohan and Kristian Stokke. Participatory development and empowerment: The dangers of localism. *Third World Quarterly*, 21(2):247–268, April 2000. ISSN 1360-2241. doi: 10.1080/01436590050004346. URL <http://dx.doi.org/10.1080/01436590050004346>. 9.3
- [82] Dominik Moritz, Bill Howe, and Jeffrey Heer. Falcon. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 1–11. ACM, May 2019. doi: 10.1145/3290605.3300924. URL <https://doi.org/10.1145/3290605.3300924>. 3
- [83] Jamie R. Nuñez, Christopher R. Anderton, and Ryan S. Renslow. Optimizing colormaps with consideration for color vision deficiency to enable accurate interpretation of scientific data. *PLOS ONE*, 13(7):e0199239, August 2018. ISSN 1932-6203. doi: 10.1371/journal.pone.0199239. URL <https://doi.org/10.1371/journal.pone.0199239>. 2.2.3
- [84] CA Okoro, ND Hollis, AC Cyrus, and S. Griffin-Blake. Prevalence of disabilities and health care access by disability status and type among adults — united states, 2016. *Centers for Disease Control and Prevention MMWR*, 67:882–887, 2018. doi: 10.15585/mmwr.mm6732a3. 1.1
- [85] Manuel M. Oliveira. Towards More Accessible Visualizations for Color-Vision-Deficient Individuals. *Computing in Science Engineering*, 15(5):80–87, 9 2013. doi: 10.1109/MCSE.2013.113. Conference Name: Computing in Science Engineering. 2.2.3
- [86] Fabio Pittarello and Manuel Semenzato. Towards a data physicalization toolkit for non-sighted users. In *2024 IEEE 21st Consumer Communications & Networking Conference (CCNC)*, pages 1–6. IEEE, January 2024. doi: 10.1109/ccnc51664.2024.10454861. URL <https://doi.org/10.1109/ccnc51664.2024.10454861>. 2.1.2

- [87] Christopher Power, André Freire, Helen Petrie, and David Swallow. Guidelines are only half of the story: Accessibility problems encountered by blind users on the web. In *Proceedings of the SIGCHI conference on human factors in computing systems*, CHI '12, pages 433–442, New York, NY, USA, 2012. Association for Computing Machinery. doi: 10.1145/2207676.2207736. 3
- [88] Blake Ellis Reid. The curb-cut effect and the perils of accessibility without disability. *SSRN Electronic Journal*, 2022. doi: 10.2139/ssrn.4262991. URL <https://doi.org/10.2139/ssrn.4262991>. 1.1, 5.6
- [89] Yvonne Rogers, Jeni Paay, Margot Brereton, Kate L. Vaisutis, Gary Marsden, and Frank Vetere. Never too old. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3913–3922. ACM, April 2014. doi: 10.1145/2556288.2557184. URL <https://doi.org/10.1145/2556288.2557184>. 5.3.3, 5.4.2.1
- [90] Rod D Roscoe, Erin K Chiou, and Abigail R Wooldridge, editors. *Advancing diversity, inclusion, and social justice through human systems engineering*. CRC Press, London, England, December 2019. 1.1
- [91] Heather Ruland Staines. Digital open annotation with hypothesis: Supplying the missing capability of the web. *Journal of Scholarly Publishing*, 49(3):345–365, April 2018. ISSN 1710-1166. doi: 10.3138/jsp.49.3.04. URL <http://dx.doi.org/10.3138/jsp.49.3.04>. 9.3
- [92] Manaswi Saha, Michael Saugstad, Hanuma Teja Maddali, Aileen Zeng, Ryan Holland, Steven Bower, Aditya Dash, Sage Chen, Anthony Li, Kotaro Hara, and Jon Froehlich. Project sidewalk: A web-based crowdsourcing tool for collecting sidewalk accessibility data at scale. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–14. ACM, May 2019. doi: 10.1145/3290605.3300292. URL <http://dx.doi.org/10.1145/3290605.3300292>. 9.3
- [93] Antti Salovaara. Inventing new uses for tools: A cognitive foundation for studies on appropriation. *Human Technology: An Interdisciplinary Journal on Humans in ICT Environments*, 4(2):209–228, November 2008. ISSN 1795-6889. doi: 10.17011/ht/urn.200811065856. URL <https://doi.org/10.17011/ht/urn.200811065856>. 2.1.2
- [94] Elizabeth B.-N. Sanders and Pieter Jan Stappers. Probes, toolkits and prototypes: three approaches to making in codesigning. *CoDesign*, 10(1):5–14, January 2014. doi: 10.1080/15710882.2014.888183. URL <https://doi.org/10.1080/15710882.2014.888183>. 2.1.2
- [95] Zhanna Sarsenbayeva, Niels van Berkel, Eduardo Velloso, Jorge Goncalves, and Vassilis Kostakos. Methodological standards in accessibility research on motor impairments: A survey. *ACM Computing Surveys*, 55(7):1–35, July 2023. doi: 10.1145/3543509. URL <https://doi.org/10.1145/3543509>. 5.3.3
- [96] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):341–350, January 2017. doi: 10.1109/tvcg.2016.2599030. URL

<https://doi.org/10.1109/tvcg.2016.2599030>. 1.2, 2.2.1, 5.3.2.2, 5.3.3, 5.4.3.1, 5.5.2.1

- [97] Anastasia Schaadhardt, Alexis Hiniker, and Jacob O. Wobbrock. Understanding Blind Screen-Reader Users’ Experiences of Digital Artboards. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI ’21, New York, NY, USA, 5 2021. Association for Computing Machinery. Accessed: 2021-09-06. 2.2.3
- [98] D. Schepers and F. Elavsky. “accessible data viz”. *The Investigative Reporters and Editors Journal*, 4:8–9, December 2024. 2.2.3
- [99] William Schiff and Emerson Foulke, editors. *Tactual Perception*. Cambridge University Press, Cambridge, England, March 1982. 2.2.3
- [100] Hanieh Shakeri, Ye Yuan, Benett Axtell, Denise Y. Geiskkovitch, and Carman Neustaedter. Designing smart home technology for passive co-presence over distance. In *Designing Interactive Systems Conference*, DIS ’24, page 3389–3406. ACM, July 2024. doi: 10.1145/3643834.3661508. URL <http://dx.doi.org/10.1145/3643834.3661508>. 9.3
- [101] Ather Sharif and Babak Forouraghi. evographs — a jquery plugin to create web accessible graphs. In *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–4. IEEE, January 2018. doi: 10.1109/ccnc.2018.8319239. URL <https://doi.org/10.1109/ccnc.2018.8319239>. 5.3.1.1
- [102] Ather Sharif, Sanjana Shivani Chintalapati, Jacob O. Wobbrock, and Katharina Reinecke. Understanding screen-reader users’ experiences with online data visualizations. In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS ’21, pages 1–16. ACM, October 2021. doi: 10.1145/3441852.3471202. URL <https://doi.org/10.1145/3441852.3471202>. 2.2.3, 5.2, 5.3.1.2
- [103] Ather Sharif, Olivia H. Wang, Alida T. Muongchan, Katharina Reinecke, and Jacob O. Wobbrock. Voxlens: Making online data visualizations accessible with an interactive javascript plug-in. In *CHI Conference on Human Factors in Computing Systems*, pages 1–19. ACM, April 2022. doi: 10.1145/3491102.3517431. URL <https://doi.org/10.1145/3491102.3517431>. 5.3.1.1
- [104] Ather Sharif, Joo Gyeong Kim, Jessie Zijia Xu, and Jacob O. Wobbrock. Understanding and reducing the challenges faced by creators of accessible online data visualizations. In *The 26th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS ’24, pages 1–20. ACM, October 2024. doi: 10.1145/3663548.3675625. URL <https://doi.org/10.1145/3663548.3675625>. 2.2.3, 3
- [105] Lei Shi, Idan Zelzer, Catherine Feng, and Shiri Azenkot. Tickers and talker. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4896–4907, New York, NY, USA, May 2016. ACM. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858507. URL <https://doi.org/10.1145/2858036.2858507>. Accessed: 2021-09-03. 2.1.2
- [106] Ben Shneiderman. The eyes have it: A task by data type taxonomy for informa-

- tion visualizations. In *The Craft of Information Visualization*, pages 364–371. Elsevier, 2003. ISBN 9781558609150. doi: 10.1016/b978-155860915-0/50046-9. URL <https://doi.org/10.1016/b978-155860915-0/50046-9>. 5.5.2
- [107] Alexandru-Ionut Siean and Radu-Daniel Vatavu. Wearable interactions for users with motor impairments: Systematic review, inventory, and research implications. In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility*, pages 1–15. ACM, October 2021. doi: 10.1145/3441852.3471212. URL <https://doi.org/10.1145/3441852.3471212>. 5.3.3
- [108] Volker Sorge. Polyfilling accessible chemistry diagrams. In *Lecture Notes in Computer Science*, pages 43–50. Springer International Publishing, 2016. ISBN 9783319412634. doi: 10.1007/978-3-319-41264-1_6. URL https://doi.org/10.1007/978-3-319-41264-1_6. 2.2.3, 5.2, 5.3.1.1, 5.3.2.1
- [109] Laura South and Michelle Borkin. Generating Seizure-Inducing Sequences with Interactive Visualizations. In *Proc. Of the IEEE VIS 2020 posters*, 10 2020. doi: 10.31219/osf.io/85gwy. 2.2.3
- [110] Laura South, David Saffo, and Michelle Borkin. Detecting and Defending Against Seizure-Inducing GIFs in Social Media. Technical report, OSF Preprints, 1 2021. Accessed: 2021-08-20. 2.2.3
- [111] Katta Spiel, Kathrin Gerling, Cynthia L. Bennett, Emeline Brulé, Rua M. Williams, Jennifer Rode, and Jennifer Mankoff. Nothing about us without us. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI ’20, pages 1–8. ACM, April 2020. doi: 10.1145/3334480.3375150. URL <https://doi.org/10.1145/3334480.3375150>. 1.1, 9.1
- [112] Clay Spinuzzi. The methodology of participatory design. *Technical communication*, 52(2):163–174, 2005. 2.1.1
- [113] Sebastian Paul Suggate. Beyond self-report: Measuring visual, auditory, and tactile mental imagery using a mental comparison task. *Behavior Research Methods*, 56(8):8658–8676, September 2024. ISSN 1554-3528. doi: 10.3758/s13428-024-02496-z. URL <https://doi.org/10.3758/s13428-024-02496-z>. 1.1
- [114] Sarit Felicia Anais Szpiro, Shafeka Hashash, Yuhang Zhao, and Shiri Azenkot. How people with low vision access computing devices. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS ’16, pages 171–180, New York, NY, USA, October 2016. ACM. ISBN 9781450341240. doi: 10.1145/2982142.2982168. URL <https://doi.org/10.1145/2982142.2982168>. 2.2.3
- [115] Pierre Tchounikine. Designing for appropriation: A theoretical account. *Human-Computer Interaction*, 32(4):155–195, July 2017. ISSN 1532-7051. doi: 10.1080/07370024.2016.1203263. URL <https://doi.org/10.1080/07370024.2016.1203263>. 2.1.2
- [116] Yuanyang (YY) Teng and Darren Gergle. Access is not enough: Toward developmental flourishing. In *Proceedings of the Extended Abstracts of the 2026 CHI Conference*

- on *Human Factors in Computing Systems*, CHI EA '26, page 1–6. ACM, April 2026. doi: 10.1145/3772363.3798552. URL <http://dx.doi.org/10.1145/3772363.3798552>. 1.1
- [117] John R Thompson, Jesse J Martinez, Alper Sarikaya, Edward Cutrell, and Bongshin Lee. Chart reader: Accessible visualization experiences designed with screen reader users. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–18. ACM, April 2023. doi: 10.1145/3544548.3581186. URL <https://doi.org/10.1145/3544548.3581186>. 5.2, 5.3.1.1, 5.4.1.1, 5.5.3
 - [118] Alexandra To, Angela D. R. Smith, Dilruba Showkat, Adinawa Adjagbodjou, and Christina Harrington. Flourishing in the everyday: Moving beyond damage-centered design in hci for bipoc communities. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, DIS '23, pages 917–933. ACM, July 2023. doi: 10.1145/3563657.3596057. URL <https://doi.org/10.1145/3563657.3596057>. 2.1.1
 - [119] Visa. Visa Chart Components. <https://github.com/visa/visa-chart-components>, 2022. Accessed: 2022-12-01. 5.3.2.1
 - [120] W3C. Web content accessibility guidelines (wcag) 2.1, 2018. 2.2.2.2, 3
 - [121] WAI. Understanding success criterion 2.4.7: focus-visible. WCAG standard, W3C, 2016. Accessed: 2022-12-11. 5.4.3.2
 - [122] WAI. Understanding success criterion 4.1.2: name, role, value. WCAG standard, W3C, 2016. Accessed: 2022-03-04. 2.2.2.2, 5.3.1.3
 - [123] WAI. Accessible rich internet applications (WAI-ARIA 1.1). Technical report, W3C, 2017. Accessed: 2022-12-11. 3
 - [124] WAI. Understanding success criterion 2.1.1: keyboard. WCAG standard, W3C, 2017. URL <https://www.w3.org/WAI/WCAG21/Understanding/keyboard.html>. Accessed: 2026-03-20, <https://www.w3.org/WAI/WCAG21/Understanding/keyboard.html>. 5.4.2.2
 - [125] Chris Weaver. Multidimensional visual analysis using cross-filtered views. In *2008 IEEE Symposium on Visual Analytics Science and Technology*, pages 163–170. IEEE, October 2008. doi: 10.1109/vast.2008.4677370. URL <https://doi.org/10.1109/vast.2008.4677370>. 2.2.1, 3
 - [126] WebAIM. WAVE, the web accessibility evaluation tool. <https://wave.webaim.org/>, 2020. Accessed: 2023-01-10. 5.5.1.1
 - [127] WebAIM. Webaim: The WebAIM Million - An annual accessibility analysis of the top 1,000,000 home pages. <https://webaim.org/projects/million/#wcag>, 2021. Accessed: 2021-09-03. 9.3
 - [128] David Gray Widder and Richmond Wong. Thinking upstream: Ethics and policy opportunities in ai supply chains, 2023. URL <https://arxiv.org/abs/2303.07529>. 1.2
 - [129] David Gray Widder, Meredith Whittaker, and Sarah Myers West. Why ‘open’ ai systems are actually closed, and why this matters. *Nature*, 635(8040):827–833, November 2024.

ISSN 1476-4687. doi: 10.1038/s41586-024-08141-1. URL <https://doi.org/10.1038/s41586-024-08141-1>. 1.2

- [130] B. L. Wimer, L. South, K. Wu, D. A. Szafr, M. A. Borkin, and R. A. Metoyer. Beyond vision impairments: Redefining the scope of accessible data representations. *IEEE Transactions on Visualization and Computer Graphics*, 30(12):7619–7636, December 2024. 2.2.1, 2.2.3
- [131] Langdon Winner. Do artifacts have politics? In *Computer Ethics*, pages 177–192. Routledge, May 2017. ISBN 9781315259697. doi: 10.4324/9781315259697-21. URL <https://doi.org/10.4324/9781315259697-21>. 1.2
- [132] J. O. Wobbrock, S. K. Kane, K. Z. Gajos, S. Harada, and J. Froehlich. Ability-based design: Concept, principles and examples. *ACM Transactions on Accessible Computing*, 3(3):1–27, April 2011. doi: 10.1145/1952383.1952384. 1.2, 2.1.1
- [133] Jacob O. Wobbrock, Shaun K. Kane, Krzysztof Z. Gajos, Susumu Harada, and Jon Froehlich. Ability-based design. *ACM Transactions on Accessible Computing*, 3(3):1–27, April 2011. doi: 10.1145/1952383.1952384. URL <https://doi.org/10.1145/1952383.1952384>. 5.6
- [134] Mintesnot Woldeamanuel and Andrew Kent. Measuring walk access to transit in terms of sidewalk availability, quality, and connectivity. *Journal of Urban Planning and Development*, 142(2), June 2016. ISSN 1943-5444. doi: 10.1061/(asce)up.1943-5444.0000296. URL [http://dx.doi.org/10.1061/\(ASCE\)UP.1943-5444.0000296](http://dx.doi.org/10.1061/(ASCE)UP.1943-5444.0000296). 9.3
- [135] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics*, 22(1): 649–658, January 2016. ISSN 1077-2626. doi: 10.1109/tvcg.2015.2467191. URL <https://doi.org/10.1109/tvcg.2015.2467191>. 2.2.1
- [136] R. Wood et al. “creatures of habit”: influential factors to the adoption of computer personalization and accessibility settings. *Universal Access in the Information Society*, 23(2): 927–953, March 2023. 3
- [137] Keke Wu, Emma Petersen, Tahmina Ahmad, David Burlinson, Shea Tanis, and Danielle Albers Szafr. Understanding Data Accessibility for People with Intellectual and Developmental Disabilities. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI ’21, pages 1–16, New York, NY, USA, 5 2021. Association for Computing Machinery. doi: 10.1145/3411764.3445743. Accessed: 2021-09-03. 2.2.3
- [138] Katherine Ye, Wode Ni, Max Krieger, Dor Ma’ayan, Jenna Wise, Jonathan Aldrich, Joshua Sunshine, and Keenan Crane. Penrose. *ACM Transactions on Graphics*, 39(4), August 2020. doi: 10.1145/3386569.3392375. URL <https://doi.org/10.1145/3386569.3392375>. 5.5.3.1
- [139] Rebecca Yeo and Karen Moore. Including disabled people in poverty reduction work: “nothing about us, without us”. *World Development*, 31(3):571–590, March 2003. ISSN 0305-750X. doi: 10.1016/s0305-750x(02)00218-8. URL [https://doi.org/10.1016/s0305-750x\(02\)00218-8](https://doi.org/10.1016/s0305-750x(02)00218-8).

1016/s0305-750x(02)00218-8. 1.1

- [140] Haixia Zhao, Catherine Plaisant, Ben Shneiderman, and Jonathan Lazar. Data sonification for users with visual impairment. *ACM Transactions on Computer-Human Interaction*, 15(1):1–28, May 2008. ISSN 1073-0516. doi: 10.1145/1352782.1352786. URL <https://doi.org/10.1145/1352782.1352786>. 2.2.3
- [141] Jonathan Zong, Crystal Lee, Alan Lundgard, JiWoong Jang, Daniel Hajas, and Arvind Satyanarayan. Rich screen reader experiences for accessible data visualization. *Computer Graphics Forum*, 41(3):15–27, June 2022. doi: 10.1111/cgf.14519. URL <https://doi.org/10.1111/cgf.14519>. 2.2.2, 3, 5.2, 5.3.1.1, 5.3.1.3, 5.3.2.1, 5.4.1.1, 5.4.3.3, 5.5.1.1, 5.5.2, 5.5.3, 5.6