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:

- Table of contents
- Chapter 1: Introduction
- Chapter 6: *Skeleton*: Visual Authoring of Non-visual Data Experiences
- Chapter 8: *Softerware*: Enabling Personalization of Interactive Data Representations for Users with Disabilities
- Chapter 9: Discussion & Future Work (Sections 9.1 and 9.3 only)
- References

This document does not contain the following chapters:

- Chapter 2: Background & Related Work
- Chapter 3: Overview of Contributions
- Chapter 4: *Chartability*: Heuristics as a Tool and Resource
- Chapter 5: *Data Navigator*: Low-level Tooling for Creating Navigable Data Structures
- Chapter 7: *Cross-perception*: Rethinking Input Design Towards Blind Analytical Interaction
- Chapter 10: Biographical Sketch

Contents

List of Figures

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" [26]. 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, 30, 99]. 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 [75] 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 [24]. 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 [100]. 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 [47]? 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 [68]: 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 [68, 80, 82, 98, 120]. 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 [113] and are sometimes, for this reason, regulated or made proprietary and controlled by powerful entities [32, 109].

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 people are *able* to do [114] and also rules and guardrails about what anyone downstream from that tool's design *should* do [32, 108]. 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 [29].

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 [85], 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 [40].

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.

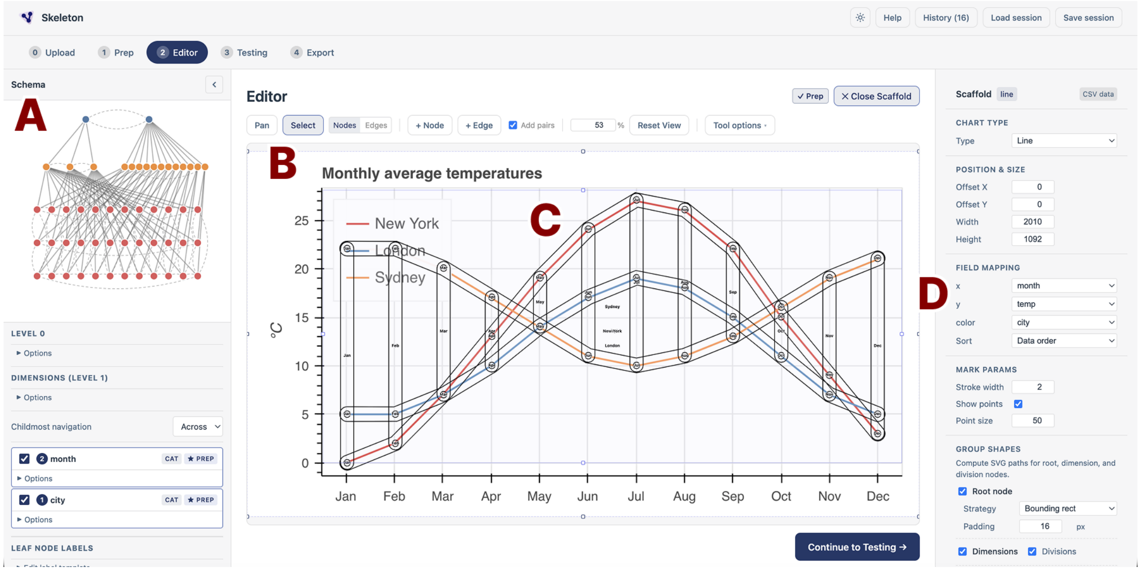
Part III

Navigation: Making Data Structures Traversable

Chapter 6
Skeleton: Visual Authoring of Non-visual Data Experiences

This chapter was adapted from my paper, currently under review with IEEE VIS:

> F. Elavsky, C. Nnadozie, L. Nadolskis, P. Carrington, and D. Moritz, '*Skeleton*: Visual Authoring of Non-visual Data Experiences', *IEEE Transactions on Visualization and Computer Graphics*, 2026.



Childmost navigation: Across

☑ 2 month CAT ★ PREP

☑ 1 city CAT ★ PREP

☑ Add pairs 53 %

Type: Line

Offset X: 0

Offset Y: 0

Width: 2010

Height: 1092

x: month

y: temp

color: city

Sort: Data order

Stroke width: 2

Show points: ☑

Point size: 50

Compute SVG paths for root, dimension, and division nodes.

☑ Root node

Strategy: Bounding rect

Padding: 16 px

☑ Dimensions ☑ Divisions

Figure 6.1: Low-fidelity design draft of *Skeleton*'s main user interface components and interactions. A. *Skeleton*, our graphical user interface for creating and debugging screen reader navigation experiences of data visualizations. B. Users can add nodes wherever they want over the chart, manually or automatically with algorithmic assistance. C. Users can then "draw" edges between nodes, which signify navigation paths through the visualization.

6.1 Abstract

When sighted practitioners author accessible data visualizations, they build navigation structures (the nodes, edges, and input bindings that govern how assistive technologies traverse an interface) entirely in code, with no visual representation. This invisibility makes navigation structures difficult to inspect, debug, and iterate on. To sighted practitioners, every other aspect of a visualization is iterated on because it is visible; navigation structure ships as a first draft, if at all,

because it is not. Without a representation to react to, practitioners cannot develop judgment about what makes navigation good or bad, and the quality ceiling of non-visual experiences is set by the absence of a feedback loop. We address this problem through longitudinal co-design with practitioners across cartography, design systems, and open-source visualization, and make three contributions. First, we introduce technical advancements for making the properties of accessible navigation structure visible and directly manipulable during authoring, grounded in two foundational pieces of infrastructure produced by our co-design work: an *Inspector* that renders navigation graphs as interactive node-link diagrams, and a *Dimensions API* that expresses navigation in terms of data dimensions rather than explicit graph construction. Second, building on these, we present *Skeleton*, a direct-manipulation authoring environment in which the properties of an accessible navigation structure are translated into visual representations authors can observe and manipulate. Key techniques include a dual-view editor that simultaneously shows the system's navigation model and the end user's spatial experience, a scaffolding engine that automates spatial node placement by repurposing a visualization rendering pipeline, a live label-template editor with real-time screen-reader-output preview, and a testing mode that makes traversal sequence visually trackable. Third, we evaluate *Skeleton* through an in-situ study with 8 practitioners across visualization design, engineering, and research. Making navigation structure visible changed how practitioners engaged with accessible design: they reconsidered the architecture of their own visualizations, attended to a broader range of input modalities, and shifted from treating accessibility as a compliance task to treating it as a design problem.

6.2 Overview

We start this work with a provocation: How might the discipline of visualization help the discipline of accessibility?

Visualization has spent decades developing techniques for a specific class of problem: representing and interacting with information visually. Grammars of visual encoding [71, 85, 107, 110, 123], direct-interaction interfaces [22, 48, 94], and iterative feedback loops between representation and understanding [27, 39, 61] are all methods that enable abstraction and manipulation of the information that underlies the visual representation. We argue these methods have a direct application within the discipline's own accessibility challenges, one that has not yet been explored.

Sighted practitioners who build accessible data visualizations face an unusual authoring problem. The non-visual navigation structures they construct (the nodes, edges, focus states, input bindings, and semantics that govern how assistive technologies traverse a chart) exist only as code. A practitioner can write a navigation hierarchy, but cannot see it, click on a node to inspect what will be announced, observe the spatial relationship between a navigation path and the chart it overlays, and then manipulate its properties through direct interaction. Every other aspect of a visualization has a visible, inspectable representation during authoring: the visual encodings are visible, the layout is visible, the interaction states are visible. And yet, navigation structure is not.

This invisibility has practical consequences. Without a way to see what they are building, sighted practitioners cannot easily catch structural errors, compare design alternatives, and iterate. Accessibility becomes downstream of every other design choice, not because practitioners choose to deprioritize it, but because the authoring conditions do not support anything else [52, 93]. The floor and ceiling of non-visual data experiences are constrained by what sighted authors can perceive of their own work.

We engage this space with the following research questions:

R1 (Qualitative, Exploratory): What challenges do sighted practitioners face when designing and engineering navigation structures for accessible visualizations?
R2 (Qualitative, Exploratory): How do sighted authors reason about the non-visual experiences that accompany their visualizations?
R3 (System, Design): How can we make the properties of accessible navigation structure visible and directly manipulable during authoring?
R4 (Qualitative): In what ways does a directly manipulable visual representation of navigation structure change how practitioners find errors and improve upon their designs?

We address these questions through longitudinal co-design with practitioners across cartography, design systems, and open-source visualization, following an action research orientation [38] in which the research team was embedded in each community's active development work. This paper makes three contributions:

First, **we introduce technical advancements** for making the properties of accessible navigation structure visible and directly manipulable during authoring. These techniques are grounded in our co-design collaborations, which produced two foundational pieces of infrastructure: an *Inspector* that renders any navigation graph as an interactive node-link diagram, and a *Dimensions API* that formalizes a declarative grammar for expressing navigation in terms of data dimensions rather than explicit graph construction (**R1**, **R2**, **R3**).

Second, building on this infrastructure, **we present *Skeleton*, a direct-manipulation authoring environment** in which the topology, spatial mapping, semantics, and input logic of an accessible navigation structure are made visible and directly manipulable. *Skeleton* is built on Data Navigator [21], a code-based library for constructing interactive data navigation structures (**R3**). Our intention with *Skeleton* is to continue to develop it towards a usable, practical system beyond the scope of this research.

Third, **we contribute findings from an in-situ interview study** with 8 practitioners across visualization design, engineering, and research. We evaluate Skeleton as a design probe [33, 49] rather than a deployable system, focusing on qualitative shifts in practitioner engagement rather than task performance. We find that making navigation structure visible shifted how participants engaged with accessible design: they reconsidered the architecture of their own visualizations, attended to a broader range of input modalities, and shifted from treating accessibility as a compliance task to treating it as a design problem (**R1**, **R2**, **R4**).

6.3 Related Work

6.3.1 Non-visual Data Experiences

Blind people who rely on assistive technologies interact with data in fundamentally different ways than sighted users who use a direct pointer, like a mouse [58, 69, 91]. A substantial body of research has documented what these experiences look like across modalities, and what it takes to make them meaningful. In the context of "data experiences," this paper focuses specifically on interactive navigation structures, but we briefly survey adjacent modalities to situate our contribution.

Alternative text and natural language. A dominant strand of this work concerns the generation and evaluation of textual descriptions of visualizations [53, 55, 59, 63]. More recent LLM-driven systems and Q/A approaches can caption charts with varying degrees of semantic depth, some at a risk of producing bias [19, 25, 57]. While alt text makes a visualization's message available without sight, it is by nature static: a description conveys what a visualization says, but not how a user might explore or interact with it.

Sonification, haptics, and tactile rendering. Non-visual data experiences extend well beyond text. Sonification encodes data as sound [9, 42, 67], with declarative grammars emerging for authoring these experiences [56]. Haptic and tactile representations offer another channel through refreshable displays, 3D-printed graphics, and multimodal touchscreen interactions [11, 45, 70, 81]. Recent systems integrate multiple non-visual modalities around a single data representation [15, 46, 88].

Interactive navigation structures. The primary focus of our work centers on the state of research related to structured navigation: the traversal of data points, groupings, and interface elements through assistive technologies and keyboard input [21, 96, 105, 122]. Existing systems and interfaces have demonstrated that going beyond static descriptions to support hierarchical, traversable data structures meaningfully improves how blind users can explore and reason about charts [101, 122]. Giving users control over the textual tokens surfaced during navigation improves comprehension and agency [51]. And more recent work has found that perceptually congruent navigation structures for charts and diagrams can improve goal-driven exploration [72].

6.3.2 Authoring Non-visual Data Experiences

Accessible visualization has historically centered the experiences of disabled users, but a parallel and increasingly urgent body of work examines the experiences of the people who build these experiences: visualization designers, engineers, and researchers.

Practitioner challenges. Research consistently finds that sighted visualization practitioners struggle with accessibility [24, 52, 93]. Most visualizations in the wild are inaccessible and designers themselves report lacking guidance, especially for complex and interactive graphics [52, 93]. And screen reader users experience the downstream effects of these gaps: inconsistent structure, poor keyboard support, and information that is present visually but absent in the accessibility tree [24, 91]. The pattern across this work is consistent: the practitioners who build visualizations lack the tools and feedback mechanisms to make non-visual experiences effective, useful, and good.

Across practitioner-centered literature, a recurring finding is that non-visual experiences are treated as downstream of visual design choices, added after the visual representation is finished rather than designed in parallel [52, 64, 93, 124]. This sequencing has consequences: what is navigable and how it is structured is constrained by whatever visual decisions came first.

Authoring-oriented systems. A distinct line of work has focused on building authoring tools and libraries that give practitioners more tractable paths to accessible output. Few visualization tools support the kinds of interactive navigation structures that assistive technology users most benefit from [59]. Of those that do, most rely on code-based approaches [6, 21, 90, 92]. *Umwelt* [124] takes a different and notable approach: it is a structured editing environment where authors specify representations *across* modalities (sonification and visualization) in an integrated interface, where navigation is made available over the visualization using *Olli* [6]. The latest work in this space is *Arboretum*, a tool that provides automatic conversion of diagrams to a tabular structure, navigable structure, and tactile representation [112]. In *Arboretum*, the input visual is treated as the ground truth for the navigation structure, which is treated as downstream output.

Communicating visually, authoring invisibly. There is a revealing irony across this body of related work: research about navigation structures almost invariably communicates those structures visually. Papers such as "rich screen reader experiences" [122], *ChartReader* [101], *Benthic* [72], and *Data Navigator* [21] each use visual node-link diagrams and hierarchical schematics to explain navigation paths to their readers. The same pattern holds in adjacent domains that involve structuring data for navigation, such as PDF remediation [74].

And in authoring-oriented systems, none provide a visual interface through which practitioners can interactively inspect and manipulate navigation structures as a first-class design material. Structure output is either downstream of code or static visuals. Structure is always *derived* or *specified*; indirectly manipulated. Additionally, verification of the structure across all of these systems requires developers to manually navigate using a screen reader after the structure has been authored and rendered, before returning to the upstream visual design space or code.

Navigation structure is understood and communicated visually by sighted researchers and designers, yet built entirely without visual feedback by developers. We seek to address this gap.

6.4 Co-design Foundations

Our research follows an *action research* orientation [38], in which knowledge is generated by engaging with a community to solve a real problem in-situ, alongside them rather than studying it from the outside. These collaborations started from a shared motivation: practitioners needed to make their existing systems accessible to navigational assistive technologies, using Data Navigator [21] as a foundation. Across three projects, we worked with 12 individuals outside our research team. Three blind co-designers shaped the work throughout: CD1 and CD2 (anonymous) and a co-author on this paper, [REDACTED]. CD1's contributions are noted in Section 6.4.1, CD2's are noted in Section 6.5.3. [REDACTED] contributed to early ideation, problem formation, and framing for the project as a whole, helping define the authoring challenges that *Skeleton* addresses, in addition to feedback on study design.

The co-design literature on accessibility has centered people with disabilities as primary design partners [16, 18, 78, 101, 121, 122], and rightly so. Our co-design takes sighted practitioners as primary partners because the authoring challenges we address happen on the sighted side of the process: it is sighted authors who cannot see what they are building.

Two themes converged across all three engagements, and we present them here before describing the individual projects that surfaced them. First, **practitioners communicated and reasoned about accessible navigation using visual representations**: while designing for language, sound, and structure, our collaborators drew on paper, built wireframes of nodes and edges, and reflected on the design space using visual artifacts. Even when collaborating with blind co-designers, a visual medium was the first language of sighted authors. Second, **development that followed visual design work faced severe iteration barriers**: verifying a navigation experience required building a working code prototype and manually navigating it with a screen reader. The gap between visual design and code-based scaffolding with manual testing produced repeated mistakes, misinterpretations, and abandoned prototypes. Each project below contributed distinct evidence for these themes and surfaced specific requirements that shaped our infrastructure and tooling.

6.4.1 Geologic Map

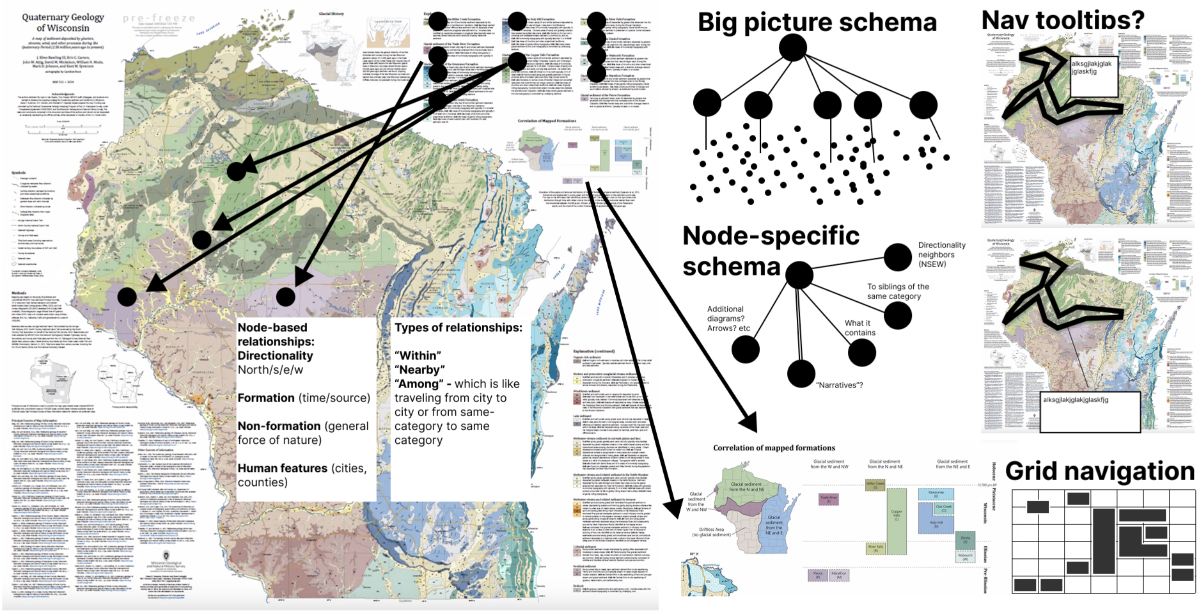


Figure 6.2: Our visual design work in Figma over a static geologic infographic map of Wisconsin. We use visual forms and illustrations over and beside the map to communicate flows, structure, navigation styles, and interaction patterns.

Our longest engagement spanned just over two years, with a cartographer, accessibility consultant, and blind co-designer (CD1) building an interactive version of a quaternary geologic map of Wisconsin [79]. The map combined dozens of irregular geographic regions organized categorically by a legend.

Early design sessions took place in Figma (Figure 6.2), where we laid out node-link diagrams of how a screen reader user might traverse the map, legend, and peripheral information. We annotated each node with text a screen reader would announce and connected them to relevant regions. Drawing the structure made it possible to discuss design choices, catch dead ends, and debate traversal strategies. We were informally doing what *Skeleton* later formalized.

The friction began when we moved from design to implementation. We had no way to verify that our designed structure would be good or ideal, and scaffolding the project into Data Navigator was arduous: the translation from even simple navigational designs to functional code was too complex to hand off or meaningfully iterate on. The collaboration also surfaced a design-space boundary: for within-map spatial navigation, an egocentric audio-game approach [5] fit better than a node-link graph, a distinction CD1 helped surface. Reaching this boundary was only possible because we could see the structure we were trying to build.

6.4.2 Design System Library

Our second engagement, spanning 7 months, was with 6 engineers and designers on Adobe's React Spectrum Charts library, an open-source chart component system. We worked in their codebase and in Miro on navigation design for bar charts, clustered and stacked bars, line charts, and related types. Because we needed generalized, reusable patterns, our design problems differed from the geologic map: we needed to account for use, re-use, and edge cases across chart types.

Our Miro sessions produced two kinds of artifacts: diagrams of navigation structure for specific chart *instances* and *schema* diagrams capturing generalized patterns in dimensional terms. The concept of a "dimension" emerged naturally: common transformations on Data Navigator's graph structure corresponded to properties within a dataset, such as a "categorical" dimension or a "numerical" dimension, where traversal took place within collections of grouped siblings. This dimensional thinking directly foreshadowed the Dimensions API (Section 6.4.4).

The dominant friction was iteration speed. We could sketch and converge on a schema in Miro in an hour, but verifying the design in a functional example required embedding Data Navigator into a large codebase, implementing changes, rebuilding, and manually testing with a screen reader. The collaboration also surfaced a limitation: mobile screen reader navigation uses swipe gestures rather than keyboard input, and our keyboard interaction model did not account for it. Together, these gaps made two requirements clear: a higher-level navigation abstraction and a visual tool for inspecting structures without a screen reader or a fully integrated build.

6.4.3 Open Source Visualization Library

Our third collaboration, spanning nearly two years, was with Quansight Labs and contributors to Bokeh, a Python-based open-source visualization library. We performed an accessibility audit [23] based on Chartability [20] and identified that Bokeh visualizations with interactive chart elements needed to be navigable by assistive technologies.

Unlike Adobe, Bokeh has no native chart types. Its API operates at the level of glyphs, renderers, and data sources, which users assemble freely. There was no standard unit around which to anchor a navigation pattern, and any grammar or tooling would need to accommodate an open-ended range of encoding combinations. The iteration gap from Adobe was present in a more severe form: making library-wide contributions to a fully open-source project required incremental tooling for the most common cases. For Bokeh, we needed to test functional, data-driven navigation abstractions without fully embedding Data Navigator into the library.

6.4.4 Infrastructure from Practice

The three collaborations converged on two concrete requirements: a way to visually render and inspect navigation structures, and a higher-level abstraction for specifying navigation without hand-wiring every node and edge. We built two pieces of infrastructure to address these requirements. Both were used in subsequent co-design work and became the foundation on which *Skeleton* was built.

6.4.4.1 An Inspector Gadget

We built an *Inspector* (`@data-navigator/inspector`) to render any Data Navigator structure as an interactive node-link graph using D3, with an accompanying console for debugging. Hierarchical structures are colored by level; edges are drawn as directed links; the entry point is visually marked. The *Inspector*'s graph can itself be navigated using Data Navigator, with visual focus tracking during navigation, allowing practitioners to manually verify structure and reachability.

This made structural verification immediate: a practitioner could generate a structure and check at a glance whether the hierarchy had the right levels, whether circular extents produced expected wrap-around edges, or whether a particular path was reachable. The interactive console logs API information and underlying data when nodes are activated, and hovering or focusing logged information highlights the corresponding node in the graph.

The *Inspector* remains a developer tool, however. It requires code familiarity to attach and renders structure as an abstract graph with no connection to the spatial layout of the underlying visualization. It shows topology but not instantiated geometry: a practitioner can see that two nodes are connected but not where their focus indicators will appear on-screen. This gap between navigable structure and its spatial instantiation over a rendered chart motivated *Skeleton*.

6.4.4.2 Alternative Dimensions

The original Data Navigator library requires explicit graph construction: practitioners specify nodes, edges, and navigation rules by hand. This is general but scales poorly.

The *Dimensions API* introduces a declarative abstraction one level above this. Rather than specifying the graph directly, a practitioner describes the *dimensions* of their data, the meaningful axes along which a user might want to navigate, and the API constructs the full node-link structure automatically. A dimension has a type (`categorical` or `numerical`), a data key, and behavioral properties governing traversal. Two properties are central: `extents` determines boundary behavior (`terminal` stops at edges; `circular` wraps around), and `childmostNavigation` determines whether leaf-level nodes are reachable laterally across a dimension's divisions without first returning to a parent.

The generated structure is a multi-level hierarchy: each dimension produces a root node, below which division nodes group the data, below which leaf nodes represent individual data points. Multiple dimensions over the same dataset share leaf nodes, so users navigating via different dimensions reach the same data through different paths. With the Dimensions API, bar chart navigation that would otherwise require constructing every node and edge by hand is expressed as a single dimension declaration:

```
dimensions: {
  values: [
    {
      dimensionKey: 'month',
      type: 'categorical',
      behavior: { extents: 'circular' }
    }
  ]
}
```

The abstraction is chart-type-agnostic: bar charts, scatter plots, line charts, and layered charts all use the same vocabulary, with different combinations producing different navigation topologies. This directly addressed Bokeh's problem: the API mirrors data fields and encoding choices as a set of dimensions.

6.5 *Skeleton*: System Design

With a visual structure renderer (the Inspector) and a declarative abstraction for producing navigation structures (the Dimensions API), the remaining problem was to bring these capabilities into an integrated, direct-manipulation authoring environment where practitioners could not only see structure but manipulate it interactively, test it with real input, and iterate on it with immediate feedback. *Skeleton* is that environment. It integrates the Inspector and Dimensions API into a unified workflow and extends both with a guided preparation phase, a spatial rendering canvas, and a live testing mode. Each of Skeleton's authoring techniques makes visible a specific property of non-visual interaction that sighted practitioners otherwise cannot see during authoring: the topology of what is navigable, the spatial mapping of where navigation lives over the chart, the semantics of what is announced, and the temporal sequence in which a user encounters nodes.

Where the Inspector requires a practitioner to write code first and visualize after, *Skeleton* reverses the direction: practitioners design a navigation structure visually and inspect the code representation as a consequence of that design.

6.5.1 Staging: Input and Preparation for Authoring

The authoring workflow proceeds through four stages: upload, prepare, edit, and test. Making a visualization accessible involves decisions about what is navigable, how navigation is triggered, and where focus indicators appear in space. These decisions are related but distinct, and collapsing them into a single undifferentiated interface, as code-only workflows effectively do, makes each one harder to reason about. The stage architecture surfaces them as separate concerns. It also preserves a direct correspondence between what practitioners see in the interface and the structure of the Data Navigator API [21]: each stage maps onto a distinct layer of the API, lowering the barrier to moving from visual authoring into code when production deployment requires it.

Upload. The upload phase is deliberately permissive. Practitioners can bring a dataset, a chart image, both, or neither. When a dataset is present, *Skeleton* parses its fields and infers dimension types automatically, producing a default, starting configuration for the *Prepare* step. When only an image is present, practitioners proceed directly to editing and construct nodes and edges manually over the image. This was motivated by our geologic map co-design work: many bespoke visualizations do not have a single underlying dataset, and any tool that requires structured data as a precondition for excludes the cases that need it most. *Skeleton* can be applied to any 2D image surface, not only to visualizations in the conventional sense.

Prepare. The prepare stage addresses a hard authoring bottleneck: not placing nodes, but deciding what structure to build at all. A practitioner who has never designed a navigation structure faces an open configuration space with no obvious entry point. The prep stage presents a four-chapter Q&A wizard that moves through authoring decisions sequentially: (1) whether the chart should have a root node and what it should announce, (2) which data fields should be navigable dimensions and how those dimensions should behave at their boundaries, (3) which keyboard interactions each dimension should be assigned to, and (4) what text labels each level of the hierarchy should produce. Each chapter is accompanied by an illustrative schematic diagram of which part of the heirarchy is being edited as well as a diagram showing examples of what these decisions look like when showing on a chart. The wizard's output populates a configuration in the editor that practitioners can then inspect, refine, and revise.

6.5.2 Edit: Interacting with Topology, Layout, and Semantics

Seeing the system, seeing the experience. The editor is *Skeleton*'s primary authoring environment (**??**) and presents two interlinked representations of the same navigation structure. A schema panel shows the structure as an abstract hierarchical tree layout that makes levels and parent-child relationships immediately readable. A graph canvas shows the same structure rendered as geometric elements positioned over the uploaded chart image, representing what an end user would encounter spatially. These two representations are bidirectionally linked: selecting a node in either view propagates the selection to the other, so practitioners can simultaneously hold in mind both the abstract topology of what is navigable and the spatial instantiation of where that navigation will live. The dual-view design makes visible a real conceptual divide between the system's model of navigation and a user's experience of it, one that practitioners recognize once they can see it, even without prior vocabulary for it.

Leveraging visualization as a scaffolding engine. Manually positioning leaf nodes over each data mark is the most mechanical step in the authoring workflow, and it scales poorly with dataset size. In our early pilot sessions, actually placing nodes in the canvas space was the slowest and most tedious part of the process. To address this, *Skeleton* includes a scaffold tool (Figure 6.3) that automates spatial placement by repurposing Vega [85] as a layout engine.

The scaffold renders a Vega chart specification to a hidden, off-screen container and extracts

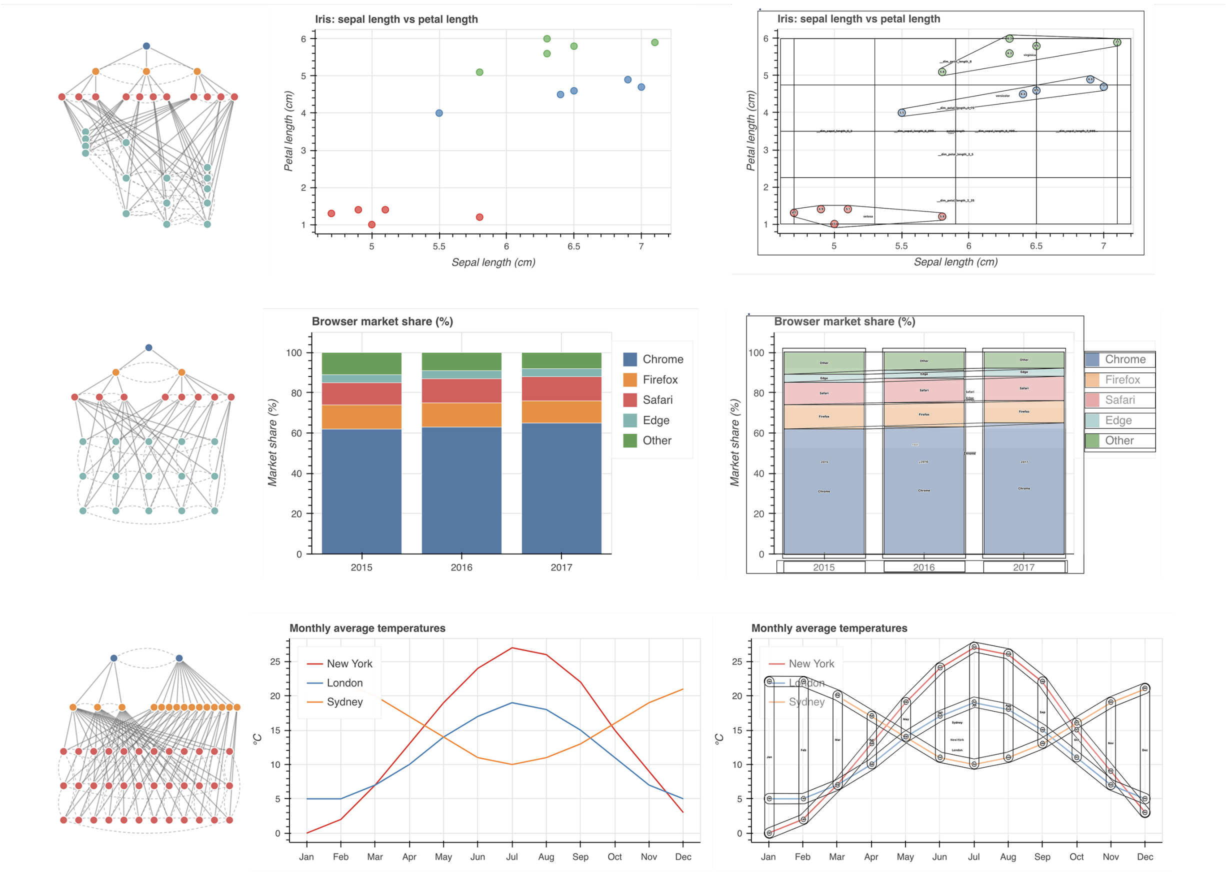


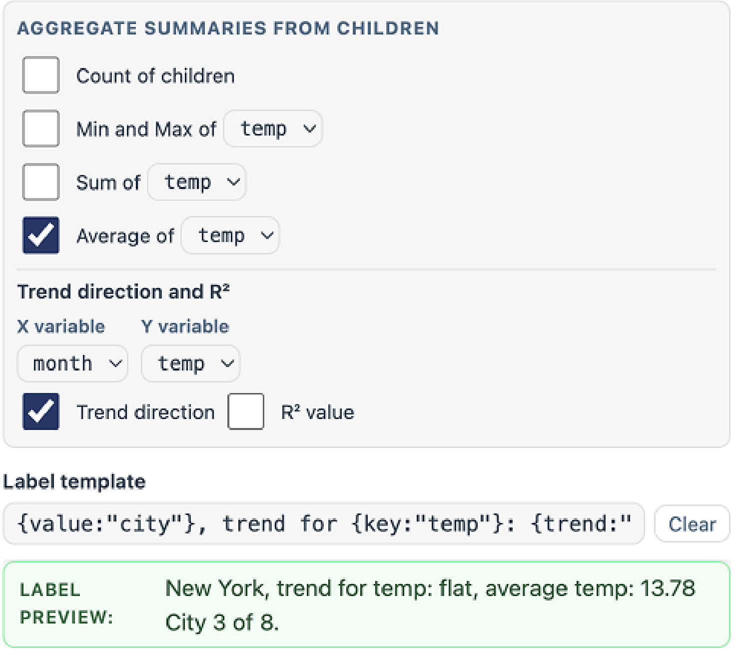
Figure 6.3: Input data transformed into a navigable structure using the *Dimensions API* and visualized with our *Inspector* gadget (left). The input chart (middle). The navigable structure is transformed and drawn over the chart using the *Scaffolding Engine* (right).

node positions via the library's internal scale and view APIs, using the rendering engine purely as a coordinate computation step: no actual chart is ever shown to the practitioner. Coincidentally, none of our co-designers were crafting visualizations using Vega or Vega-Lite (or derivatives), yet the Vega rendering engine could faithfully reconstruct every necessary mark position over the underlying, inaccessible data visualization provided to *Skeleton*.

Additionally, positions and outline strategies for category-level group nodes are computed from the leaf positions using geometric algorithms: a union of child node paths, a convex hull, a grid over numerical bounds, or a bounding rectangle. These designs were consistently produced as ideal treatments during co-design, so we chose them for our starting set of outline strategies.

The scaffold is optional and works by generating synthetic placeholder positions that practitioners can adjust manually. It dramatically improved authoring speed: in light pilot tests, a research team member completed a three-dimension structure without scaffolding in 8 minutes 22 seconds and with scaffolding in 56 seconds. A co-designer completed the task incorrectly (failing to account for one dimension's division nodes) in 13 minutes 7 seconds without scaffolding, and correctly in 2 minutes 44 seconds with it.

Specifying token patterns, editing instances. Selecting any node populates a properties panel with spatial properties (position, size, shape) and semantic properties (ARIA role, description, and a label template editor; Figure 6.4). The label template allows practitioners to assemble the text a screen reader will announce at that node from tokens drawn from data fields, including aggregate statistics at group-level nodes and precise data values at leaf nodes [51]. Editing labels can apply to all nodes of the same type or to a single instance.



AGGREGATE SUMMARIES FROM CHILDREN

☐ Count of children

☐ Min and Max of `temp`

☐ Sum of `temp`

☑ Average of `temp`

Trend direction and R^2

X variable: `month`

Y variable: `temp`

☑ Trend direction ☐ R^2 value

Label template

`{value:"city"}, trend for {key:"temp"}: {trend:"`

LABEL PREVIEW: New York, trend for temp: flat, average temp: 13.78 City 3 of 8.

Figure 6.4: Group label pattern builder, including an array of aggregate summary options, template formatter field, and preview.

At the bottom of the semantic section, a live preview displays the full assembled announcement string in the exact form a screen reader would produce: role, semantics, group membership, and label combined into a single rendered output that updates in real time. Prior to this preview, understanding what would be announced at a given node required running a screen reader and navigating to it sequentially. In deeply nested structures, this meant spending several seconds listening to labels and drilling in. This was consistently the main point where bugs were produced and missed during our co-design work.

The preview makes text announcements inspectable as visible, editable objects. This surfaces a class of highly specific, low-level problems that code-only workflows leave invisible: redundancies in announced text, missing contextual framing, and label ordering and punctuation that affects comprehension and reading speed. Of all the authoring decisions *Skeleton* exposes, label templates involve the most degrees of freedom and have the most direct bearing on the quality of the non-visual experience. Getting the structure right ensures navigability; getting the labels right determines whether navigation communicates anything meaningful.

6.5.3 Test: Debugging Interaction Interactively

The testing stage allows practitioners to navigate the structure they have built using the same keyboard input and navigation rules that assistive technologies would use, without leaving the tool. When a practitioner enters testing, the three Data Navigator modules are instantiated in sequence: the structure module rebuilds the navigation graph from the current configuration, the rendering module creates an HTML layer positioned over the chart image at each node's spatial coordinates, and the input module registers keyboard listeners for all navigation rules. The result is a live, keyboard-navigable structure. An event log records navigation events in order, letting practitioners verify that all nodes are reachable and that label sequences make sense when encountered serially rather than read simultaneously.

The abstract graph continues to display during testing. As the practitioner navigates, the focused node is highlighted simultaneously in the canvas (showing its spatial position over the chart image) and in the abstract graph (showing its structural position in the hierarchy). This parallel tracking makes the temporal traversal sequence visible, showing the order and path through which a user encounters nodes, so practitioners can verify at a glance both where focus is and what role it occupies. A text-chat mode is also available, in which practitioners navigate by typing natural language commands, motivated as a design by the Adobe collaboration (Section 6.4.2), to explore interaction alternatives for mobile screen reader users.

The testing stage also served, during development, as the primary debugging interface for *Skeleton*'s own data pipeline. Errors in position computation, label resolution, or structure generation that would propagate silently through code became visible the moment a node highlight appeared in the wrong location or a label read as undefined. A tool for making non-visual structure visible turned out to benefit from the same property during its own construction.

Practice-based validation. CD2, a subject matter expert who professionally evaluates interfaces for screen reader access and is familiar with other visualization navigation systems, used *Skeleton*'s testing stage to evaluate navigation output across several chart types and dimension configurations: line charts (3 configurations), bar charts (2), stacked bar charts (3), and scatterplots (4). This evaluation followed a manual, systematic approach combining standards-based criteria with expert screen reader testing. Scatterplots required the most iteration, surfacing bugs in Data Navigator's core library that were then fixed. CD2 also recommended that we rely on list-based navigation while in the editor (before the testing stage) in case users build themselves into a keyboard trap.

6.6 User Study

Our co-design work followed an action research orientation: the research team was embedded in practitioner communities, and our system work was motivated by the problems those communities faced. This process generated the techniques and infrastructure that comprise *Skeleton*, but we still needed to understand the impact our system had on visualization practitioners more broadly. Our co-designers were deeply familiar with the problem space, having spent months or years working on accessible navigation. We needed to understand what happens when practitioners who are *not* embedded in this process design, author, and debug navigation structure visually: whether the representations we built are legible to them, whether the techniques change how they reason about accessible design, and what new questions or problems emerge when navigation structure becomes visible. These are empirical questions that required a study.

To evaluate how *Skeleton* influences the way practitioners engage with accessible design, we conducted an in-situ interview study with 8 participants across visualization design, engineering, and research. The study was conducted remotely over video call, took approximately 45–60 minutes per session, and was approved by our IRB. Participants provided verbal consent at the start of each session. Video and audio were not recorded, however some participants consented to share the data/image they brought to the study as well as screenshots of their workflow; data collection was note-based throughout.

6.6.1 Participants

Participants were recruited through snowball sampling within the visualization and accessibility community, and through referrals from co-designers involved in our earlier collaborations. We asked each participant to self-report their primary work role (engineering, research, design, or student) and their existing level of accessibility expertise on a 1–5 Likert scale. We also asked whether the visualizations they build are ever bespoke, that is, custom rather than instances of a recognizable, standard chart type. This distinction mattered because bespoke visualizations represent an especially underserved case in accessible design tooling: no library pattern applies, and every navigation structure must be designed from scratch. Participants were not compensated.

6.6.2 Procedure

Each 45-minute session proceeded in four phases. Before the session, all participants were asked to prepare a chart image they were currently working on or had recently built, for use in the third phase.

Phase 1: Introduction and demographics (5 minutes). After obtaining verbal consent and recording a pseudonym, we collected self-reported role, accessibility expertise level, and whether the participant regularly builds bespoke visualizations.

Phase 2: Generic chart think-aloud [2] (10 minutes). Participants used *Skeleton* on a provided bar chart and dataset of fruit counts (Apples, Pears, Nectarines, Plums, Grapes), asked to design an accessible navigation experience for a screen reader user with no instructions on how the tool worked. The tool loaded with a default structure having both a categorical dimension (`fruit`) and a numerical dimension (`count`) active. This default was intentionally problematic for two reasons: numerical navigation sorts by count value and groups data into subdivided ranges, producing a traversal order different than the visual layout an additional, largely unhelpful level in the hierarchy for such a simple chart. We observed how participants reasoned about what they saw and whether and how they noticed this extra dimension.

Phase 3: Own-chart think-aloud (15 minutes). Participants loaded their own chart image and attempted the same task, except they were also asked to explain their graphic to the research team (purpose, role, data, and domain). This phase was open-ended: charts ranged from standard types to bespoke visualizations, and the goal was to observe how participants reasoned about navigation structure when the context was their own work.

Phase 4: Reflective interview (15 minutes). We conducted a semi-structured interview in which participants reflected on their decisions in Phase 2 versus Phase 3, their experience with the generic versus their own chart, and their assessment of the tool's capabilities and limitations. We asked what felt possible or impossible, what they wanted to do that they could not, and what they found themselves thinking about that they had not considered in Phase 2.

6.6.3 Analysis

Notes from each session were compiled into a shared document. Participant quotes reported in the results are reconstructed from these researcher notes, not verbatim transcripts. We analyzed the data using a combination of thematic analysis [8] and affinity diagramming [36], iterating across both methods to surface recurring patterns while preserving the specificity of individual participant experiences. Analysis attended particularly to differences in how participants engaged with accessible design before and after using the tool, the range of input modalities and user scenarios they considered, and moments when participants reconsidered or wished to redesign their own visualizations.

6.7 Results

We organize our findings into five themes that emerged from thematic analysis and affinity diagramming across all eight sessions. Each theme captures a qualitative pattern in how practitioners engaged with accessible navigation design when its structure was made visible and manipulable. We report these findings descriptively and ground them in specific participant moments; interpretation follows in the Discussion.

6.7.1 Seeing Navigation Made Structural Problems Legible as Design Problems

The generic bar chart in Phase 2 loaded with an intentionally problematic default: both a categorical dimension (`fruit`) and a numerical dimension (`count`) were active, producing overlapping navigation structures with different traversal orders over the same data. This configuration is a poor design choice, arguably a design failure, but one that would be difficult to detect in code alone (**R1**, **R4**).

Participants varied widely in how quickly they recognized the problem. P1 turned off the numerical dimension within seconds of seeing the editor, without commenting on it. Most participants, however, initially struggled to understand what they were seeing, remarking on the unfamiliar structure: "what is this? what are these?" when encountering the numerical divisions for the first time. P8 spent time trying to guess what the extra divisions represented but did not remove them during Phase 2, only realizing during Phase 3 that the additional dimension was "probably bad." P2 and P5 expressed suspicion early: P5 asked, "Is this too much data? This seems like way too much to just navigate through," and P2 noted, "I feel like a lot of data points would be bad, yeah? Like, too many at once is bad?" Both of these remarks were prompted by the visual density of the structure, not by navigating it.

The testing stage (Section 6.5.3) proved critical for resolution. P4, P5, and P7 each removed the extra dimension only after navigating the structure with keyboard input in testing mode, where the traversal sequence made the redundancy experientially apparent. In total, five of eight participants resolved the problem during the session: P1 and P2 during editing, and P4, P5, and P7 after testing.

Beyond the intentional default, participants identified other problems through visual inspection. P8 reacted to a generated node name: "Okay dim_fruit node... that is horrible, what is that?" During Phase 3, P7 looked at the edges of their multi-line chart and asked, "are all these bad? Is it bad that I don't even really know what the takeaway of this [structure] is?" P3, seeing a full hierarchy for their own simple six-item bar chart (during Phase 3), concluded "I should just skip the root and grouping and go straight to the data. This seems like too many steps." In each case, the visual representation of their navigation structure motivated judgment about potential negative design qualities.

6.7.2 Practitioners Developed a Designerly Interest in What Constitutes Good Navigation

The most pervasive pattern across sessions was that participants began asking design questions about navigation quality, unprompted by any instruction or guidance from the research team (**R2**). These questions went beyond identifying errors: participants wanted to know what *good* navigation would be for their charts.

P2, working through the bar chart in Phase 2, deliberated over boundary behavior: "Loop back or stop? I don't think there is a right way. I will just pick *fruit* for now and *loop* and see what this does." P6, who brought a bespoke flower visualization, wondered how to translate the affective quality of their chart: "I think my visualization should be more about the vibes, but I don't know how to make the alt text have good vibes. What is *fun*?" P4 asked fundamental questions about the interaction model itself: "Why do screen readers and keyboards have to work this way? Do people like that?... why do we navigate?" And later after testing, P4 concluded, "I bet we should make this *faster*" before cutting out the additional numerical dimension in Phase 2.

Several participants engaged with the concept of narrative and flow. P8 articulated this as a question about the goal of the visualization: "sometimes I want a big picture, not precision. I may want to drill down a little..." and observed that "we think too much in terms of components... sometimes accuracy isn't the actual goal, it's getting a general sense of something." P4, upon discovering that *Skeleton* supports text-based input, asked, "How do you make that good, though? Like chatGPT, or do people want to, like, interact with the chart [elements]?'

During the interview, several participants explicitly requested guidance. P1, P2, P3, and P6 wanted to see examples of well-designed navigation experiences. P1, P3, P4, P5, P6, and P7 wanted embedded guidelines within the tool. P2 and P4 actually used web search to look for "chart navigation for accessibility guidelines" (P2) and "accessible viz screen reader design" (P4). P3 was interested in automation and heuristics that could suggest reasonable defaults. These requests are consistent with the pattern (**R2**): practitioners who could see the design space wanted orientation within it.

6.7.3 Iteration Was Substantive, Self-directed, and Concentrated on Semantics

Every participant iterated on their navigation designs, and this iteration was neither perfunctory nor prompted by the research team (**R4**). Participants revisited decisions, revised configurations, and in some cases restructured their entire approach after encountering their design in a new stage of the tool.

The most sustained iteration concerned labels and text announcements. Every participant spent the largest share of their authoring time in the label template editor (Figure 6.4), editing the text that a screen reader would announce at each node. This editing was granular: participants wrote full sentences, rearranged the order of data tokens, debated whether to include field keys alongside values or values alone, experimented with how to name groups and individual elements, and considered the length and density of the resulting announcements. At the division and dimension levels, some participants added multiple aggregate statistics (count, sum, average, range, trend) and then returned to trim them. P2, for instance, edited data point labels, left them, and returned to revise them two additional times: "This label is way too complicated, I think." These label iterations were small, fast, and frequent, and they reflected an intuitive grasp of the importance of the textual tokens that constitute a screen reader user's primary interface to data.

A second, distinct pattern of iteration emerged around testing. The editor displays all navigation nodes simultaneously, showing the full structure at a glance. The testing stage (Section 6.5.3), by contrast, shows only the currently focused node, highlighting it in the canvas one at a time as the practitioner navigates. This difference consistently prompted participants to revise. Several returned to the editor after testing to adjust group node outlines, because outline strategies that looked distinct when displayed simultaneously (such as bounding rectangles vs. convex hulls) became harder to differentiate when encountered one at a time. Others adjusted their dimension configurations: P3 and P5 restructured their dimensions after testing, and P4, P6, and P8 experimented with different key bindings and navigation rules. P2 used the testing stage specifically to identify labels that needed revision at the dimension and division levels. In these cases, testing was not only treated as a bug-finding activity but also as a way to encounter, reason about, and then improve a design that had looked adequate in the editor but felt inadequate in sequential traversal.

The most extensive iteration came from P1, who returned to the preparation wizard after reaching the testing stage and re-took the entire wizard from scratch. Seeing the full structure in testing motivated them to re-evaluate their earliest decisions. They reduced their navigation from three dimensions to one, producing a navigation experience they described as deliberately minimal. In the reflective interview, P1 explained that they had been thinking about trimming excess, joking that they were "working on a data-to-word ratio."

The scaffolding tool shaped the pace and character of these iterations. Participants who used the scaffold (Figure 6.3) generally required only minor manual adjustment of node positions, spending one to two minutes refining placements in-aggregate before moving on. Iteration on spatial layout was primarily about the appearance of group-level outlines rather than individual node positions: because the scaffold computed leaf positions from the chart's encoding, the main remaining spatial decision was how division and dimension nodes should visually indicate their grouping. When participants returned to scaffolding, it was most often because they wanted to change the chart type used for coordinate computation or to try a different group outline strategy.

6.7.4 Seeing Navigation Prompted Practitioners to Reconsider the Architecture of Their Own Charts

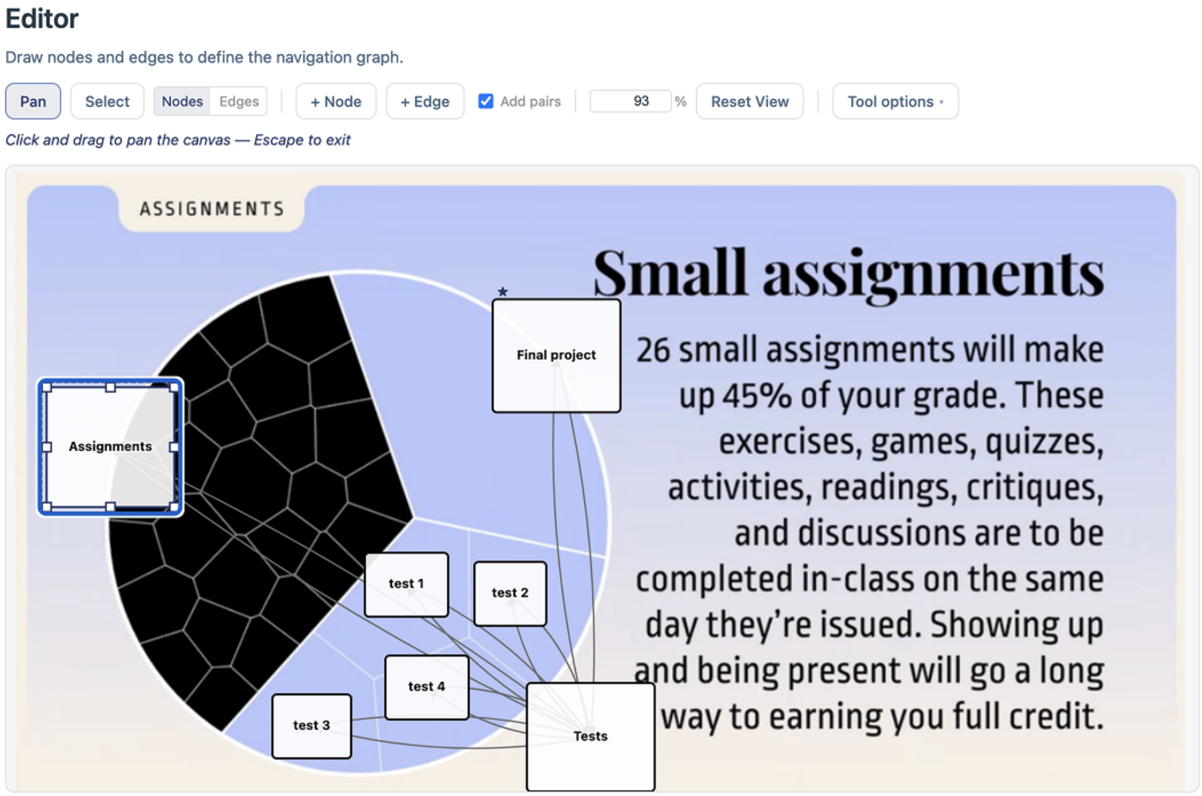
An unexpected finding was that working with *Skeleton* prompted several participants to question the design of the visualization they had brought, not just the navigation structure they were building over it (**R4**). Five participants (P4, P5, P6, P7, P8) expressed a desire to simplify their own charts after seeing what the corresponding navigation structure required. Three (P1, P5, P6) considered whether a different chart type might serve the same communicative goal with a simpler navigational architecture. And in 2 cases (P1, P6), participants questioned whether a chart was the right medium at all.

P5, who brought a scatter plot with over two thousand data points, initially tried to build a navigation structure over the full dataset, recognized that the result was unmanageable, and then asked: "do I even need visualization?" They went on to wonder whether the information could be communicated as "a few sentences or like, some data [users] can prompt" (referring to using a large-language model). P6, who brought a bespoke flower visualization in which each petal encoded a different variable, initially tried to express the chart's full complexity in navigation (grouping by flower and then navigating by petal) but then reversed course: "okay, what if we actually just treat this like a bar chart?" Using the scaffold tool, they produced a simple list-style navigation that worked well for their data, and reflected: "my visualization has a lot, but actually, this could be pretty easy to navigate, I think." P6 also wondered whether there was a non-chart way to "tell their story," and in further discussion described something closer to a scrollytelling article or guided walkthrough than a single interactive chart.

P8's case was the most striking. They brought a voronoi pie chart (Figure 6.5) used to communicate to students that 26 assignments make up 45% of their grade, the largest slice of the pie. Working through *Skeleton*, P8 first considered making all 26 voronoi cells navigable, then considered making only the three main slices of the pie navigable, and then questioned whether navigation was needed at all. The "point of this," as they described it, was to communicate a single ratio; students did not need to traverse individual cells to understand it. P8 concluded that well-written alternative text was sufficient for the screen reader experience and chose not to build a navigation structure. They also reflected on the design of the visualization itself, concluding that the voronoi treatment served a visual purpose (it is visually striking) that was separable from the informational purpose (communicating a grade breakdown). They kept the visual design and simplified the non-visual experience accordingly.

6.7.5 Experiencing Keyboard Navigation Surfaced a Broader Range of Users and Input Technologies

The testing stage (Section 6.5.3) provided what was, for most participants, their first experience of keyboard navigation through a data visualization. Only three participants (P1, P2, P8) reported prior experience using a screen reader, and only two (P1, P8) had previously navigated a visualization using keyboard input alone. Yet during the study, every participant navigated using



Draw nodes and edges to define the navigation graph.

Click and drag to pan the canvas — Escape to exit

26 small assignments will make up 45% of your grade. These exercises, games, quizzes, activities, readings, critiques, and discussions are to be completed in-class on the same day they're issued. Showing up and being present will go a long way to earning you full credit.

Figure 6.5: Re-creation of P8's moment of realization, placing nodes manually: not every element in their voronoi pie chart *should* be navigable.

a keyboard.

Several participants responded to this experience with genuine engagement. P4 reacted with enthusiasm: "Woah, this is incredible. Wait, what other visualizations can do this?" and later, "Maybe you could make a visualization game, where you can move around the data?" P7 said, "I love this. I want to make charts with this." P5, who had not previously encountered keyboard-navigable charts, said, "I didn't know you could do this." These reactions were not about the tool; they were about the interaction modality itself, about the experience of traversing data spatially using directional input.

The experience also expanded participants' sense of who these structures serve (**R2**, **R4**). P8 adjusted a node's spatial dimensions and said, "Let's make the hitbox as big as possible here…for my neighbor with Parkinson's to click these," treating the navigation structure as relevant to motor accessibility, not only screen reader access. P3, observing the visual focus indicators that appeared during scaffold-based placement, remarked that "this is for more people than [someone] blind," recognizing that sighted keyboard users and users with partial vision also rely on visible focus states. During the interview, P4 asked sustained questions about what kinds of technologies different people with disabilities use, and P4 and P7 both asked why focus indication was designed to be visible rather than invisible.

P4's curiosity extended to alternative input modalities. Upon learning that *Skeleton* supports text-based navigation and (due to leveraging Data Navigator) also supports a wide array of other input modalities, they asked, “Can you talk at it, too? Is that what some people do?” and followed up with, “How cool is that? How do you make that good, though?” P8 raised a concern about discoverability in the current interaction model: “I’ve never used j or w to drill out, always shift arrow or option arrow…” and worried that a user might not know how to exit a nested level. These observations reflect an broad mental model of the user, from an abstract “screen reader user” to a person with multiple capabilities, preferences, and interaction patterns (**R2**).

6.8 Discussion

6.8.1 Visibility as a Precondition for Iteration

The central finding of this work is not that *Skeleton* taught sighted practitioners how to design accessible navigation, but that it gave them something to react to. When navigation structure was invisible, our co-designers would only seek to verify whether navigation followed our design documentation. Once visible, questions shifted to whether it was good, why, how it could be improved, and what else it could do. Visibility encouraged iteration, and iteration is enabled continuous design improvements.

This mechanism is consistent with Schön’s account of reflective practice [87]: designers iterate by externalizing a representation, perceiving its properties, and responding to what they see. The representation talks back, and the designer adjusts. But this loop requires a representation. In our co-design work, we assumed that the gap was hand-off between design and development, and the slow process of manual verification. Instead, the gap is that development was not participating in, and encouraging, further design reflection. In a developer’s code-only workflow, navigation structure has no externalization that supports this kind of perceptual engagement. A practitioner can read the code that specifies a navigation graph, but they cannot see the graph, cannot perceive its topology at a glance, cannot notice that a label is redundant or that a hierarchy is too deep by looking at it. The reflective loop is disrupted and occluded at the first step.

Skeleton encourages this loop by rendering navigation structure in a form that supports the same kind of perceptual judgment that visualization practitioners already apply to every other aspect of their work. The results show what happened when this loop was available: every participant iterated, substantively and self-directedly (Section 6.7.3). They revised labels repeatedly, restructured dimensions after testing, adjusted group outlines when sequential traversal revealed problems that simultaneous display had hidden. P1 restarted the entire preparation wizard after seeing their structure in testing mode. These are not the behaviors of practitioners following a specification; they are the behaviors of practitioners negotiating with a design material.

The co-design work reported in Section 6.4 provides complementary evidence: in each collaboration, sighted practitioners naturally reached for visual representations when reasoning about navigation, through node-link diagrams in Figma, schema sketches in Miro, and annotated wireframes on paper. They were already thinking visually about non-visual structure; the development tooling simply had not caught up. This has a practical implication for the broader field: if sighted authors depend on visibility, then any authoring workflow that keeps navigation structure invisible limits design iteration. Making structure visible does not guarantee good design, but it is a precondition for the kind of sustained, judgment-driven refinement that good design requires.

6.8.2 From Compliance to Design

A consistent pattern across the accessibility literature is that practitioners frame accessibility as a compliance problem: a set of requirements to satisfy, a checklist to complete, a legal or institutional obligation to meet [20, 52, 93]. The framing matters because compliance and design orient practitioners toward fundamentally different activities. Compliance asks: "does this pass?" Design asks: "is this good?" Our results suggest that *Skeleton* produced a partial shift from the first orientation to the second. Participants' initial instincts clustered around compliance-oriented responses: provide alternative text, follow guidelines, ask an expert. But alongside that desire for guidance, they began doing something compliance framing does not typically produce: they encountered complexity and then *iterated*. They revised labels repeatedly, restructured dimensions after testing, debated boundary behavior and hierarchical depth. This sustained, self-directed refinement is the behavioral signature of design, not compliance. As P4 put it: "I'd love to have someone blind actually just with me while I make this, but I also understand that I should learn what makes a good experience too."

The shift extended beyond the non-visual experience itself. As reported in Section 6.7.4, five participants reconsidered the design of the visualization they had brought, not just the navigation structure overlaid on it. Making the accessibility consequences of visual design choices visible prompted practitioners to question whether a different chart type, a simpler encoding, or a non-chart medium might better serve their communicative goals. This interrelation between non-visual and visual design suggests that the widespread treatment of accessibility as a compliance activity may be partly a consequence of tooling that offers no legible, manipulable design surface. Auditing frameworks are valuable, but they are *evaluative* tools, not *authoring* tools. The field needs both.

6.8.3 Bespoke Visualizations as an Unaddressed Accessibility Research Problem

Diagrams, infographics, and data-driven illustrations are often one-off, custom designs with bespoke symbols, layouts, and visual languages. These representations are increasingly common in journalism, scientific communication, personal projects, art, and public-facing data work, and they represent the cases where accessibility-focused tools are needed most and available least. *Skeleton*'s image-based workflow (upload any 2D image, place nodes manually) provides a starting point, but the study made clear that bespoke visualizations need more than node placement. They need support for reasoning about what navigational structure (if any) is appropriate when no template applies, a research problem that remains largely unexplored.

6.8.4 What Visualization Owes Accessibility

Our approach, to make visual non-visual experiences, should not be limited to data visualization's own accessibility challenges. Navigation structure is a foundational component of accessible experience across domains: PDF and document reading order, web page structures, and software application layouts. In each of these areas, sighted practitioners author non-visual experiences without visual feedback, and in each, the same gap between design intent and verifiable outcome constrains quality. Testing is slow, error prone, and requires expertise in assistive technology use. Visual tooling for authoring, inspecting, and debugging non-visual structure (**R3**) is a tractable and high-value problem across application domains.

There is also a deeper question about what visibility can accomplish in principle. Work on multi-modal authoring environments has argued for de-centering visual representation and treating modalities as equal partners in the design process [124], an important ethical commitment. *Skeleton* does not do this: it re-centers visual representation as the medium through which sighted practitioners engage with non-visual structure. We believe this is justified pragmatically, because sighted authors need to articulate navigation design in their own perceptual language before they can reason about it at all, and this paper provides evidence that they do. But articulating a design in one's own language is not the same as understanding how it will be experienced in someone else's. The structures participants built during our study were never evaluated by blind users, and visibility alone cannot substitute for that evaluation. The risk we want to name is that making non-visual structure visible to sighted practitioners could be mistaken for making it *understood*, when in fact it makes it *designable*, a real but bounded gain. The fuller design process requires collaboration with disabled users, not as an occasional supplement but as a regular practice. *Skeleton* can make that collaboration more productive by giving both parties a shared representation or space of translation between representations, but it cannot replace it.

What visualization owes accessibility, then, is not simply better output but authoring tools that better stimulate reasoning, both individually and collaborative, about design.

6.9 Limitations and Future Work

Skeleton makes navigation structure visible to sighted authors, but it cannot reassure those authors whether the structure they have built is good for the people who will use it. The tool surfaces design questions; it does not answer them. Several participants asked what constitutes good navigation, and *Skeleton* had nothing authoritative to offer. Our study with sighted practitioners evaluated whether making navigation structure visible stimulated design consideration, not whether the designs sighted practitioners produced were actually good. These are related but distinct questions, and the second remains open. CD2's expert screen reader evaluation of *Skeleton*'s navigation output (Section 6.5.3) provided practice-based validation for several common chart types and surfaced concrete bugs, but this evaluation was neither comprehensive nor controlled: many configurations remain untested, and expert review is not a substitute for evaluation with a broader population of end users.

Additionally, we treated *Skeleton* as a design probe [33, 49] rather than comparing it to a controlled condition: the goal was to elicit qualitative insight about how the tool elicits engagement, not to measure performance differences.

Our approach using *Skeleton* as a design probe only with sighted participants is both a limitation and, we believe, the right sequencing: *Skeleton* improves the intentionality and iterability of what sighted practitioners produce, which is a necessary precondition for a subsequent evaluation or future collaborative design work between sighted and blind authors. Further research and evaluation should close this loop, ideally to engage how mixed-ability teams co-design multi-modal data experiences.

Skeleton is also a prototype with substantial work remaining. Not within the scope of the paper, but our participants provided ample feedback on the functionality of the prototype itself. The most urgent gap is export functionality: practitioners can design and inspect navigation structures in the tool but cannot yet produce deployable output.

6.10 Conclusion

Accessible navigation structure has long occupied an awkward position in visualization practice: known to matter, difficult to design, and invisible to the people responsible for building it. The invisibility was not incidental. Without a way to see what they were making, sighted practitioners could not catch errors, could not iterate, and could not develop the kind of considered judgment that good design requires. Accessibility remained downstream of every other decision not because of any single failure, but because the authoring conditions did not support anything else.

Skeleton demonstrates that those conditions can be changed. Making navigation structure visible and manipulable, as an interactive graph rendered over the spatial layout of a real visualization with live label previews and testable traversal, shifted how practitioners engaged with and reasoned about accessible design. They began asking qualitatively different questions than someone seeking compliance: whether the features and design of their structure was good, despite not having readily available answers.

If any conclusive take-aways can be gleaned from this project: for researchers interested in engaging accessibility, this would mean future projects might explore translational spaces between visuals and non-visuals that help sighted partners engage blind designers. For practitioners who build, design, or audit this would mean that more visualization is needed in current tooling, to enhance and make multi-modal existing non-visual methods of authoring and evaluation.

What the paper leaves open is more than what it closes. We used *Skeleton* as a design probe with sighted practitioners; we did not evaluate the navigation structures they produced with the end users those structures are meant to serve. And the broader disciplinary conversation, about what visualization research owes accessibility and what methods might transfer between them, has more questions than answers. We offer *Skeleton* not as a solution to these problems but as evidence that engaging them directly, with the full weight of visualization's methodological tradition, is both possible and fruitful.

Part V

Personalization: System-building for User Agency

Chapter 8
Softerware: Enabling Personalization of Interactive Data Representations for Users with Disabilities

---

This chapter was adapted from my published paper:

> F. Elavsky, M. Vindedal, T. Gies, P. Carrington, D. Moritz, and Ø. Moseng, 'Towards *softer*ware: Enabling personalization of interactive data representations for users with disabilities', *Computer Graphics and Applications*, 2025 (to appear at *IEEE VIS 2026*).

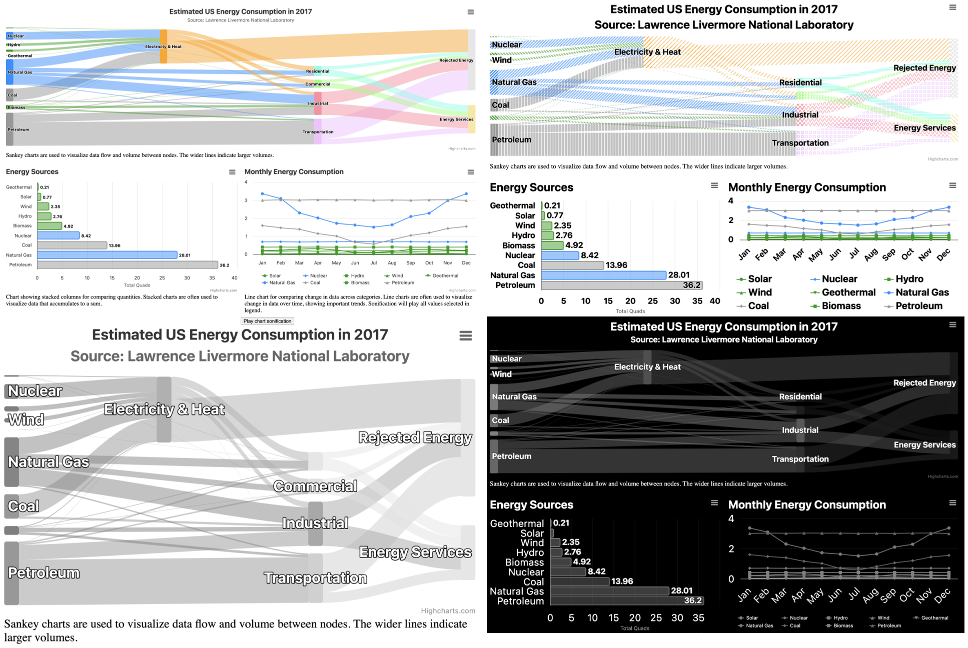
8.1 Abstract

Accessible design for some may still produce barriers for others. This tension, called access friction, creates challenges for both designers and end-users with disabilities. To address this, we present the concept of softerware, a system design approach that provides end users with agency to meaningfully customize and adapt interfaces to their needs. To apply softerware to visualization, we assembled 195 data visualization customization options centered on the barriers we expect users with disabilities will experience. We built a prototype that applies a subset of these options and interviewed practitioners for feedback. Lastly, we conducted a design probe study with blind and low vision accessibility professionals to learn more about their challenges and visions for softerware. We observed access frictions between our participant's designs and they expressed that for softerware's success, current and future systems must be designed with accessible defaults, interoperability, persistence, and respect for a user's perceived effort-to-outcome ratio.

8.2 Overview

There is a significant and relatively unacknowledged problem in emerging work on accessible visualization: a single design cannot satisfy all users. People with disabilities, even those who share the same category of disability, often have different experiences, capabilities, and needs. As experienced practitioners and researchers who have been working to make data visualizations more accessible (some of us for more than a decade), we have each observed this persistent problem in our own practice.

Data visualizations that are produced by a designer for an audience tend to be designed in a way that is relatively *unchangeable*. As a material, we use the metaphor that the creator of a visualization manipulated their design while it was in a *softer* state, like clay. And eventually, the clay is *hardened* into a state that is presented to the user. Often visualization design artifacts cannot easily be altered by an end-user after they are created. Pixels cannot be moved, graphics



Sankey charts are used to visualize data flow and volume between nodes. The wider lines indicate larger volumes.

Figure 8.1: Sometimes one design is not enough. Our design (upper left) and three different designs by low vision users. All low vision users chose larger text, but then diverged: redundant-encoding enabled (upper right), high zoom and greyscale on white (bottom left), and then dark mode (enabled externally) with greyscale (bottom right).

cannot be re-embedded. The clay has been fired and the visualization is now *baked*. We intend to make visualizations easier to change and manipulate for end-users. We want a material that is softer than software. But we also don't want a fully malleable interface, like in end-user programming, either. We propose to call this space of design "softerware."

We wanted to explore material softness *with constraints*. We conjecture that advancements in malleable interfaces and end-user programming are too system-centric and open-ended. End-user programming puts too much burden on end-users to know the language and symbols of interface-building in order to build their own interfaces. Instead, we chose to enable end-users to have agency over a data visualization interface by exposing a preferences-driven menu of options built from our existing knowledge of visualization and accessibility.

Exploring the *softerware* gap in data visualization became our primary research focus, which led us to formulate the following qualitative exploratory research questions:

- **R1**: What constraints and capabilities should we provide end-users to give them meaningful agency over interactive data visualizations?
- **R2**: What qualities, challenges, and design opportunities do designers and engineers envision for a data visualization *softerware* system?
- **R3**: What qualities, challenges, and design opportunities do blind and low vision users envision for a data visualization *softerware* system?

We contribute our findings from this research to the larger accessibility and visualization communities in hopes that we can inform and inspire future work that investigates *softerware* systems, end-user design, preferences-based user experiences, and fluid and malleable interfaces focused on end-users with disabilities. Our future work will focus on completing a finished version of our prototype and deploying it at scale within Highsoft's Highcharts ecosystem [44].

8.3 Related Work

Our contribution is an attempt to bridge the gap between the knowledge we have on accessibility for visualization (as a complex space of design and engineering) with research and practice that centers on users with disabilities being able to adust, change, or control the interfaces they interact with. We intend to frame our work towards the benefit of data visualization designers, system engineers, and end-users of data visualizations. We believe that more flexible data visualization systems that enable user preferences will require a careful approach to architecture and thorough consideration for the burdens placed on end-users.

8.3.1 Data Visualization and Accessibility

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 [34, 47]. 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.

In general, accessibility concerns itself with a broad spectrum of barriers that people with different disabilities face. And while most literature focuses on visual disabilities [69, 111], there are growing resources on areas such as cognitive/intellectual disabilities [116, 118], neurodivergence [102], and both research and systems exploring epilepsy and vestibular/motion inaccessibility in visualization [95, 97].

Yet despite these resources, making data visualizations more accessible remains a difficult task for practitioners [52, 93]. 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 [86] while another stresses that redundant encoding strategies are necessary [20]. Understanding how to make the correct design decisions may sometimes be impossible. Either existing guidelines are incorrect or it is possible that access friction becomes inevitable the more we know what different barriers look like for different people with disabilities.

8.3.2 Systems that Adapt

One angle of exploration that has been engaging this issue already focuses on systems that can adapt. Work on adaptive systems for people with disabilities, such as in *ability-based design* [114], stresses the importance of design alleviating burdens placed on users. Users who don't fit initial system designs are often expected to adapt to fit the system. This means that they may have to acquire an assistive technology, learn a peripheral skill, hack the system, or wait on a design fix. This places the burden on the user to fit the system. Ability-based design instead stresses that systems should be capable of automatically adapting, in order to reduce these burdens placed on the system's users.

However, building data visualizations that automatically adapt to users via some form of data collection often do so through means such as monitoring live biometric data and input patterns, collecting a user's self-declared conditions and cognitive ability, parsing a user's history, and sensing a user's environmental or situational context [119]. We argue that these methods for an adaptive system raise questions of end-user agency, trust, privacy, and awareness in regards to the system decision-making [76]. They may not be sufficient for addressing a user's needs while also preserving their privacy and agency.

8.3.3 Personalization and Accessibility

Lastly, we researched broader spaces where users have more design agency and explicit awareness of a system that is built to be adapted. We were interested in literature and projects that explore ways end-users can enact meaningful change on an interface, with special attention paid to accessibility and disability.

One specific project has emerged at the intersection of accessibility, visualization, and customization which focuses on screen reader users adjusting the content of textual tokens when navigating data visualizations [51]. While this is excellent work, we still have larger questions about when preferences, options, and customizations are appropriate and in what contexts as well as other ways of conceptualizing end-user agency over a system. It remains unclear when, why, and how customization and personalization can be used effectively when designing a system.

In the field of meta-design, meta-designers consider these end-user manipulations of a system to be one facet of "end-user design" and "continuous co-design" between a system and a user [65], which helps give us some meaningful language to refer to our system goals.

Recent work on the influential factors for personalization and adoption of accessibility settings [117] also informs our work in 2 key ways: conceptual mismatching between a system and user can contribute to a system's under-use while features that propose value, are time-saving, or reduce cognitive load for a user can contribute to positive perception and use of personalization of a system.

8.4 Presenting: *Softerware*

Softerware is a vision for software design that is not just based on giving a user the ability to set preferences or personalize. *Softerware* is about the intentional design of a software system that enables people with disabilities to have meaningful, opinionated, and persistent agency over that system.

We contribute the concept of *softerware* to the larger community of researchers and practitioners because we argue it is a useful construct that can help us categorize past work, improve existing projects, and inspire new directions. *Softerware* systems have been part of existing work for decades, but we lack a cohesive way to refer to designing and engineering experiences that enable end-users to have agency over malleable interfaces without entering into the territory of end-user programming and end-user development.

8.4.1 Defining *Softerware*'s Principles

Here we present the principles that define *softerware* before demonstrating an example instantiation in the context of online, interactive data visualization.

8.4.1.1 Principle: Has Reasoned, User-centered Constraints

An important aspect of *softerware* is that it is *softer* than software (which is already-baked) but not quite as *malleable*, free, and potentially low-level as systems that facilitate fully realized end-user programming and development [12, 60].

End-user programming is still a form of *programming* in the end. It focuses on taking constructs, functionalities, and reasoning from software programming and development and presenting these elements to users in ways that may suit a user's natural language or mental models, such as through no-code, visual-only, or low-code approaches. We anticipate that many users, especially those experiencing accessibility barriers, will have difficulty interacting with software paradigms based on end-user programming and development.

Instead *softerware* engages this limitation through reasoned constraints that leverage conceptualizations and language focused on overcoming anticipated user barriers. Providing constraints and then framing and presenting those constraints in ways that have vocabulary correspondence to user needs is what separates *softerware* from existing work and literature on end-user programming and development.

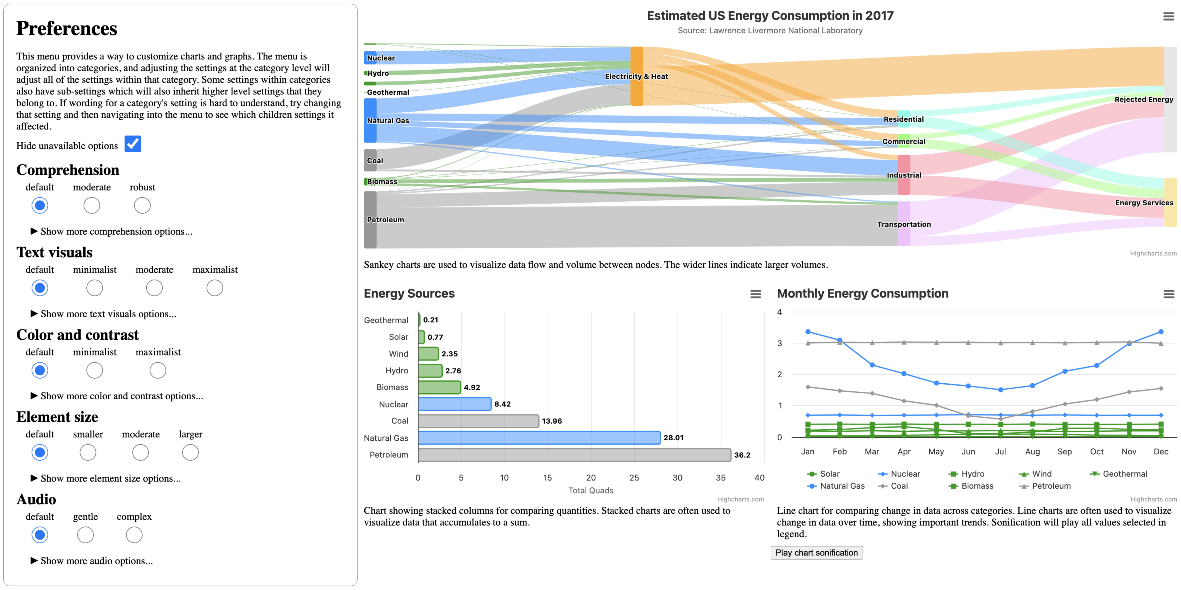
To accomplish this, the *softerware* system designer must work to anticipate not only what their system should do in a default or beginning state but also which ways that system will potentially fall short and require fitting by the end-user. The system designer should motivate all of the capabilities of a *softerware* system based on what they anticipate users will want to change, how users can discover that change is possible, and then how best to enable users to enact that change easily.

8.4.1.2 Principle: Facilitates End-user Agency

Softerware is ultimately about the process of architecting and implementing a system that enables an end-user to be able to easily express meaningful changes to that system's appearance and behavior.

Accessibility has been framed as a tension between fit and scale [43], where *fit* refers to a system that is perfectly complimentary and synchronized to a user and *scale* refers to a system that is capable of reproducing functionality for many different users. We believe that the tension between fit and scale, in addition to *access friction*, can both be alleviated when a system is designed to facilitate end-user agency.

The cornerstone goal of a *softerware* system is an attempt to facilitate *self-fitting* at a minimum, and in ideal circumstances also facilitate social methods of sharing fitting (such as loading profiles or ingesting metadata from others).



Preferences

This menu provides a way to customize charts and graphs. The menu is organized into categories, and adjusting the settings at the category level will adjust all of the settings within that category. Some settings within categories also have sub-settings which will also inherit higher level settings that they belong to. If wording for a category's setting is hard to understand, try changing that setting and then navigating into the menu to see which children settings it affected.

Hide unavailable options ☑

Comprehension

● default ○ moderate ○ robust

▶ Show more comprehension options...

Text visuals

● default ○ minimalist ○ moderate ○ maximalist

▶ Show more text visuals options...

Color and contrast

● default ○ minimalist ○ maximalist

▶ Show more color and contrast options...

Element size

● default ○ smaller ○ moderate ○ larger

▶ Show more element size options...

Audio

● default ○ gentle ○ complex

▶ Show more audio options...

Estimated US Energy Consumption in 2017

Source: Lawrence Livermore National Laboratory

Sankey charts are used to visualize data flow and volume between nodes. The wider lines indicate larger volumes.

Chart showing stacked columns for comparing quantities. Stacked charts are often used to visualize data that accumulates to a sum.

Line chart for comparing change in data across categories. Line charts are often used to visualize change in data over time, showing important trends. Sonification will play all values selected in legend.

Figure 8.2: Our dashboard design on US Energy Consumption in 2017 with a sankey, bar chart, line chart, and preferences menu on the left.

8.4.1.3 Principle: Demonstrates Value

Existing literature makes one thing particularly clear when it comes to personalization and end-user design: it has to be worth it [117]. Users must be able to recognize barriers, issues, or shortcomings of a system and then discover and utilize capabilities provided to them to eliminate or alleviate those barriers.

This entire process must not be too burdensome and the payoff should establish an expectation that future use of the system will be improved. The time and effort it takes for a user to fix a problem should be less than the time and effort generated by that problem. This means that *softerware* systems can likely be optimized and improved significantly over time, as better techniques are developed to perform tasks as quickly and easily as possible.

The user should also be able to validate the value of their interaction with a *softerware* system through the continued use of that system. If something was a painful experience and they took action to alleviate that issue, they should be able to observe the effects easily.

8.5 Prototype: Visualization *Softerware*

We first built a data visualization dashboard (see **Figure 2**) that would allow us to build a *softerware* system prototype that we could demonstrate to designers, engineers, and evaluate with end-users with disabilities. You can view and interact with our prototype, including view our open source code and dataset on user preference options, on github.

We chose a relatively clean and standard dataset that would be relevant to our target end-users, who we would recruit from the United States, on 2017 US energy consumption. This dataset afforded us enough complexity to build multiple data visualizations in one dashboard (including an uncommon type like the sankey). This choice also allowed us to explore more ideas for user interventions and investigate broader questions in our eventual study. Our dashboard was built in JavaScript and laid out in a wide format for interacting with on a desktop machine.

We designed the dashboard to contain some interactivity, but nothing highly complex. Each chart has tooltips and visual filtering provided on hover/keyboard focus and the line chart has data filtering through the legend as well as sonification.

8.5.1 *Pretty Accessible* by Default

In order to really test *access frictions*, we wanted our dashboard to be considered accessible in its default state. We ran manual and automated tests according to accessibility standards as well as a 50-heuristic manual accessibility evaluation specific to interactive data visualizations [20, 103]. In addition, we chose to use Highcharts to create the visualizations in our dashboard because existing work has demonstrated the breadth of their accessibility capabilities [59].

We ensured that text contrast was strong, text size was well above guidelines, interaction targets were of a minimum size, screen reader access was descriptive and interactive, our DOM was structured into a hierarchy, we provided semantic HTML data tables for each chart, and there were no *critical* issues detected when running automated tests using *axe DevTools* software. Many of the capabilities that made our dashboard accessible were provided out of the box by Highcharts.

8.5.2 Reasoned Constraints: 195 Accessibility Options for Interactive Data Representations

After building our initial dashboard (see Figure 2), our research team collaborated and discussed potential access frictions that might arise from the use of our dashboard. We used our existing experience from accessibility work, more than a decade in the case of the Highsoft and Elsevier co-authors, to assemble a list of concrete, expected user barriers that would be hard to resolve in a single design, and organized those barriers based on common themes.

We then used these themes to brainstorm how we anticipated end-users would identify barriers and then what language they might use to describe an identified barrier and overcome it. One example might be under the theme “hard to see”: “I can’t read this text” as a barrier with the final language for them overcoming that barrier as an expression similar to, “I wish this text size was larger” or “make this text bigger.”

From these solutions, we re-framed the language into interactive option categories. All categories were given binned options and not more than 7 choices, to avoid overwhelming users. For example, “text size” was an option category and had 7 binned choices from “small” to “large” available. We then created a hierarchy of option categories underneath these higher-level categories which could allow for more element-level specificity within a data visualization. So “text size” as a higher level category with options also contained children that had more specificity, such as “title text size,” “legend text size” and so on. The final stage of our language and options design was in line with prevailing industry wisdom, which was to avoid organizing the language of our configurations based on categories of disability [3] and instead focus on higher-level categories of identifiable elements of a user's experience, such as "text visuals," "audio," and "color and contrast."

At the end, we produced 195 option categories and 774 total option choices. Using the combinatorial rule of product, we calculated 6.83e114 possible unique end-user design configurations from these choices, which is more than the estimated number of atoms in the universe.

However, to ensure the scope of our user study was feasible, we reduced our initial working option categories down to 33 with 137 total options (and 9.35e19 possible design combinations), all focused on options we believed would be most relevant for users who are blind or low vision. These options and our subset are viewable in our live, interactive demo online.

8.5.3 Preferences Menu Design

We then iterated on visual and functional designs to allow users to actually interact with and enact these design configurations. Our early ideas included a natural language interface (since we used a relatively "natural language" centered process to develop these categories), direct manipulation of the elements in a visualization (through focus, hover, click, or selection methods), and eventually settled on the user interface of a separate, visually nearby menu with nested options (see Figure 2). We designed our menu so that manipulating higher level options in the hierarchy would enact downstream options to follow suit, but any manipulation to downstream options would override higher controls, following common patterns used in systems that implement hierarchical specificity.

We justified our user interface as a menu for our final choice because it provides a place for metadata from the other design ideas (natural language and direct manipulation) to live, in case we develop those down the line as well. We anticipated that a menu not only provides a means of interaction but also storage of the state of a system. In addition, this type of user interface is common and relatively recognizable.

8.6 Evaluation

Our first research question for this project ("What constraints and capabilities should we provide end-users to give them meaningful agency over interactive data visualizations?") focused on our thematic collation and compilation of anticipated access frictions, but our following two research questions would require outside evaluation: "What qualities, challenges, and design opportunities do designers and engineers envision for a data visualization *softerware* system?" and "What qualities, challenges, and design opportunities do blind and low vision users envision for a data visualization *softerware* system?"

8.6.1 Preliminary: Visualization Practitioners

The preliminary step in our evaluation was to investigate what qualities, challenges, and design opportunities data visualization engineers and designers envision for a *softerware* system.

8.6.1.1 Recruitment

We recruited 4 data visualization practitioners, each with roles as a current or former visualization software engineer (3) or designer (1). We recruited participants from our existing network of engineers and designers, requesting participation via email. Our practitioners were not compensated for their participation and we asked them up front if they would be willing to volunteer their time for us.

8.6.1.2 Procedure

We conducted 30-minute, semi-structured, qualitative interview sessions either over Zoom or in-person. The session consisted of a 5 minute explanation, 5 minute demo of our prototype's capabilities, and a series of open-ended, semi-structured questions for 20 minutes. Our questions started with getting their thoughts on the idea, what they anticipated other developers and designers would think, what aspects of a visualization they believe end-users will want control over, issues they believed end-users would face, and what new opportunities they envision our prototype and underlying design concept of *softerware* enables.

8.6.2 Study: Blind and Low Vision Users with Accessibility Expertise

Our primary study was focused on our third research question on the qualities, challenges, and opportunities that users with disabilities, in this case users who are blind and low vision, envision for a *softerware* experience of interactive data visualizations. To explore this, we used our prototype dashboard as a design probe to stimulate concrete feedback and ideation on both the details of our prototype as well as our larger design concept of *softerware*, borrowing from Noor Hammad's method used when exploring accessibility preferences of users in novel streaming software [33].

Our study is intended to contribute qualitative knowledge, largely because we believe that statistical generalizations or controlled experiments about a particular group or subgroup of people with disabilities may actually produce knowledge that reinforces the existing problems we are trying to address. Instead, we are explicitly interested in knowledge and experiences that might exist on the margins, even knowledge as specific as a single individual's preferences. We want to explore ways that broader guidelines and design knowledge are capable of producing artifacts that still retain barriers for some individuals with disabilities. This larger challenge (of general guidelines that do not provide a meaningful fit for individuals living with disabilities) is not new to accessibility and assistive technology research [77].

And to this end, we designed our study to maximize the production of knowledge that is considerate and careful of individual differences, challenges, preferences, and envisioned opportunities.

8.6.2.1 Recruitment

Our study involved 9 total participants who are blind or low vision (see **Table 1**), all of whom are also professionals with accessibility expertise (either currently or formerly employed in an accessibility-specific role as subject matter experts). 5 of our participants self-identified as male,

Table 8.1: Study Participants

PID	Age	Gender	Disability
P1	39	F	Totally Blind
P2	38	F	Totally Blind
P3	46	M	Legally Blind
P4	28	F	Low Vision
P5	34	M	Legally Blind
P6	52	M	Totally Blind
P7	56	F	Low Vision
P8	36	M	Low Vision
P9	55	M	Totally Blind

4 as female. Average age of our participant group was 42.67 (SD = 9.99). We initially recruited 6 participants using an existing, compensated research relationship between Highsoft and an external consultancy. In addition, we recruited 3 more participants from our existing network of accessibility consultants, who were each compensated 100 USD for their time.

We anticipated that recruiting participants who not only have lived experience with a disability but also are subject matter experts in accessibility would contribute to the depth of our qualitative study as well as general breadth of considerations. We wanted to maximize the value of feedback on our work.

We reached out to all participants via email with a call for participation and participants were screened according to whether they are blind or low vision. Participants were notified in advance of compensation and that consent to participate is voluntary.

8.6.3 Procedure

Our qualitative study sessions were recorded and conducted over zoom in 3 primary phases (plus a break) during one 90 minute session. Our phases were: early interview, task-evaluation of our dashboard (menu hidden) with discussion, a break, and task-evaluation of our dashboard (menu shown) with final discussion.

8.6.3.1 Introduction and Early Interview [20min]

Our session opened with an introduction to the research team and gathering verbal consent from participants for participation. We gathered demographic information from participants and asked them about their current assistive technology use. We followed up with questions related to whether or not they customized their technology in any way, through adjusting settings, modifications, adding scripts, getting extensions, or equivalent. We then ended the opening session with ice-breaker questions about whether they can recall a chart or graph they have experienced in the past and what their favorite way to experience a chart is.

8.6.3.2 No Menu Prototype, Tasks, Discussion [30-35min]

The next phase of our session involved showing participants our demo environment (see Figure 2), except that our preferences menu was hidden. We explained what the dashboard was and entailed, including explaining each chart type shown (sankey, bar, and line) and how to read them. We gave users a short amount of time to explore the dashboard, and then notified them that in order to evaluate the effectiveness of our technology, we would be asking them to perform 2 data tasks, one elementary and one synoptic [1]. Our intention for performing tasks was not to measure speed or accuracy of the participants, but simply as a probe for eliciting feedback on the usability and effectiveness of our prototype and design.

Our first question was to answer an elementary analytical task (direct or indirect lookup), “Does petroleum or nuclear contribute the most to Electricity and Heat?” (“nuclear” was correct). Our second was a synoptic task (pattern identification or multi-value comparison), “Which energy type has the highest use in Dec *and* Jan?” (“natural gas” was correct). We gave participants a limited time (5mins total) to answer the questions and upon answering, we gave them the correct answer and asked them to explain their process of finding their answer, step-by-step.

Our final step in this process was to interview them about their perceived challenges and frustrations with the dashboard and whether anything could be changed or adjusted in order to help them complete their tasks. We followed this phase with a 10 minute break.

8.6.3.3 Menu Prototype, Tasks, Discussion [30-35min]

We opened the final phase of our study by sending our participants a new link to a version of our online dashboard that included our preferences menu. We explained the purpose of the menu and gave them 5 minutes to explore the available options.

After participants explored the menu and its effects, we repeated our tasks procedure. Participants were given 2 tasks to complete in five minutes. First, an elementary analytical task, “Where does most coal go?” (“electricity and heat”) and then a synoptic task “In the summer, June through August, which energy type has the highest consumption rate?” (“petroleum”).

Our final discussion focused on investigating our participant’s thoughts on our prototype, the idea of preferences and customization, why they chose the customizations that they did, whether they had any new or additional ideas, considerations for other users with disabilities, and any other concerns, challenges, or feedback. We asked them specifically to consider both their personal, lived experience with their disability and assistive technology in addition to their professional expertise in accessibility.

8.7 Results

We performed two analyses from our studies, first analyzing our findings from our preliminary study from practitioners and then analyzing our results from our study with end-users. We collated our notes and transcript materials, coded them thematically, and then used affinity diagramming to group the themes that emerged from our data [35].

8.7.1 Preliminary Findings

To avoid repeating information between our preliminary study with visualization practitioners and final study with blind and low vision participants, any findings from our end-users that are echoed by our practitioners will be mentioned later. Only the findings unique to our preliminary study will be included here.

8.7.1.1 Alleviating Situational Barriers

3 of our 4 practitioners spoke about the potential benefits of end-user manipulation of a visualization for situational or contextual reasons. One participant gave the example that when giving a presentation using an existing dashboard, having the ability to manipulate features to suit layout, flow, and interactions on-demand would be valuable. Another example given was that at times a user's viewing device (such as a smartphone) can cause barriers, so softerware would be useful to have available.

8.7.1.2 Creating Potentially Harmful Visualizations

The second theme from our practitioners (3) was the concern that end-users would be able to create a misleading or harmful data visualization. For example, we have studies on how encoding area size [62] and aspect ratios [14] can be misleading or deceptive, yet being able to manipulate these for accessibility and contextual barriers (such as viewing a chart designed for desktop on a mobile phone) are important design considerations. Users may accidentally adjust features of a visualization while self-fitting that actually create problematic designs. Being able to design a system to avoid this would be important.

8.7.1.3 Designing via *Softerware*-first

The final theme from our practitioners was around authoring and design-tuning via *softerware*, where 3 participants discussed using a direct-manipulation or LLM-based *softerware* interface to author data visualizations and 1 of the 3 also mentioned that large-scale, privacy-preserving data collection from users could be used to create smarter design defaults in the future. We believe that both suggestions mirror existing work that speaks of the benefits of "design-through-use" and "continuous-co-design" [65].

8.7.2 Prototype-level Feedback

The advantage of a study with participants who had accessibility expertise in addition to their lived experience as people who are blind or low vision is that we were able to get feedback on our existing prototype as well as on our larger idea space for *softerware*.

8.7.2.1 Navigation Structure Options

While not a theme across participants, P2 mentioned that they would like to be able to navigate a data visualization using headings with their screen reader, rather than via regions. ("Regions" are a type of semantic markup used to create programmatically recognizable organization for screen readers.) This was a suggestion that immediately led to our team iterating in parallel on ideas the next day. More than 71% of screen reader users navigate information via headings when first encountering a new web page [104]. This suggestion made sense to explore as a sensible default.

8.7.2.2 Previewing Change

Our low vision users (P4, P7, P8) requested a feature that we had originally designed but not implemented, which was to directly show what different options would look like in the preferences menu itself. For example, the "text size" options would either have a preview of the text size shown for each option in the menu (like showing "Large" in the actual resulting large text size) or with a nearby preview window that would show the result of a selection as it is being selected. Low vision users in particular often use high levels of magnification and zoom, so the live results shown in the visualization space required users to go back and forth between the menu once an option was chosen and into the chart space to find what had been affected.

8.7.2.3 Language Re-consideration

Some of our participants (P1, P2, P4, P6, P7, P8) noted ambiguity or lack of clarity in the wording we used for our menu's higher level options, such as "Audio" having options for "default," "gentle," and "complex" while lower level options that inherited these were hard to connect to. For example, "Sonification order" under "Audio" had the options "default," "sequential," or "simultaneous." "Gentle" in "Audio" would set the child setting for sonification order to "sequential," but this was unclear initially.

Other participants (P1, P2, P3, P6) noted that while the menu's focus on functional categories was helpful, it might be nice to also have a way to customize the menu itself or view it from a "disability" perspective, so they could get all the screen reader options in a single place. Users were interested in looking at all options relevant to "screen readers" or "low vision" together.

8.7.3 System-class Accessibility Findings

The next set of themes that emerged were considerations that both our end-users and our visualization practitioners shared, which we are calling *system-class* considerations, using the phrase from Chris Fleizach and Jeffrey Bigham [28]. In order for *softerware* to function at scale, certain technical and infrastructure considerations would have to be prioritized to make things possible. This theme emerged thanks to the accessibility knowledge and expertise of our end-users and engineering concerns of our practitioners.

8.7.3.1 Persistence

Every participant (P1, P2, P3, P4, P5, P6, P7, P8, P9) as well as all of our practitioners noted that the ability to create some sort of "profile" or persistent state of their customizations would be one of the most important features that would make *softerware* actually useful.

> "What if I come back to this? Will I lose this? Do I need to do it again?"—P6

We followed up by asking whether certain contexts would make persistence more or less important. We asked users whether a random website or news article with a chart in it would be worth their time, to which most users replied, "no." However, P5 noted that "This is so fun that if it was there I still might play around with it and use it, especially if I had the time." P4 related this issue to an existing frustration with video games, noting that having to set up repeated options for every game was time consuming. It would be nice if they could "do this once and forget it."

8.7.3.2 Profile Sharing

Following this theme of establishing a profile, most of the participants (P1, P3, P4, P5, P6, P8) and 2 of our practitioners also expressed interest in being able to share their own profiles or ingest settings from others, in order to save others or themselves time. In phase 1 of our procedure, we asked users about their existing levels of modification, customization, and preferences setting in their existing use of technology. While all of our participants (except for P9) customize, personalize, or modify their technology to some degree, those who were most interested in customization or spoke the most about it (P3, P4, P5, P8) were also the most passionate about being able to save *other* people time and not just themselves.

> "I customize my tech a lot. If I use something for the first time and it feels off, I find a way to fix it. But most people aren't like that; it takes too long. So I love when I can share [my modifications and customizations] with others."—P3

8.7.3.3 Cross-system Interoperability

Closely related to *persistence* and *profile-sharing* was an idea expressed by several participants (P3, P4, P5, P7, P8) that they wanted to be able to use these settings outside of Highcharts and even outside of the web. "Will this work in Microsoft Excel?" and "I use salesforce for analytics a lot and would love this there," remarked P4. However, cross-system interoperability would require multiple charting libraries being intelligent enough to ingest user settings, when most are currently incapable of even recognizing a system's "high contrast" settings being active. In addition, all 4 of our practitioners suggested that there would need to be a system in place, either at the operating system level or as some kind of service hosted by a platform, where these settings could be recognized and ingested. For cross-system interoperability to be made possible, it would require establishing standards for customization, standards for preserving user privacy, and coordination with the larger community of visualization practitioners and software providers.

8.7.4 User-Centered Findings

The last major set of themes is related to the considerations of end user experience of a *softerware* system applied in practice, including our observations about the differences between users and their choices when personalizing a data visualization interface.

8.7.4.1 Frictions in User Differences

Our first major user-centered finding was that no participant chose the same set of preferences as another. Every user discussed different reasons for justifying their choice of options. Users even chose options that others specifically emphasized were inaccessible to them. An example of this was a tension in preference for and against use of "dark mode" designs.

> "If anything has dark mode? That's great. I wish everything used dark mode."—P4

P4 mentioned that they had "night blindness" (*nyctalopia*), which is why dark mode designs are helpful for them. However, P7 also mentioned that they had progressive nyctalopia, but dark mode makes an interface "virtually impossible" to them.

> "Oh, I can't use dark mode at all. I hate when websites have [dark mode] because it can be virtually impossible to use."—P7

Any one of the designs chosen by a low vision participant would have been insufficient for providing access for any of our other low vision participants (see Figure 1).

Our blind participants also had different justifications and preferences for their text and audio customizations. Some justified their differing preferences with similar justifications, such as cognitive accessibility and text description length. For example, P9 stressed that "I prefer to keep things simple" to "avoid overwhelm" while P2 said, "more information is better than less, when it comes to data." Both P9 and P2 preferred "accessible defaults," but disagreed on what length of textual descriptions should be default.

8.7.4.2 Accessible Defaults are a Necessary Prerequisite

Several participants were concerned that this approach would allow designers and developers to continue to make inaccessible charts (P2, P6, P7) if users have the ability to *self-fit*. Participants emphasized how important it is to have strong accessibility *before* customization is introduced (P2, P4, P6, P7, P9). Even 3 of our 4 of our practitioners expressed worry that *softerware* could put a design burden on users.

P9, our only participant who almost exclusively uses default settings (and avoids mods and extensions) with their current assistive tech, stressed the importance of well-thought out defaults. It is clear that for users like P9 in particular, strong defaults are much more important than customization. Although less common, some assistive technology users are not interested in the work involved in personalization and would prefer technology to suit their needs out of the box.

This leads us to argue that there is a line between ethical use of *softerware*, which is built on top of already-accessible material, and *softerware* that is filling gaps in poor design. Designs that are lacking access that wouldn't cause any friction for someone else if they were present, such as simply having alt text, aren't in need of *softerware*, they're just in need of accessibility.

8.7.4.3 Effort-to-Outcome Ratio

As a playful rephrasing of the visually-centric (and controversial) *data-to-ink* ratio [17], we observed an *effort-to-outcome* ratio among our participants, in line with previous results from existing work [117]. Nearly all participants (P1, P2, P3, P4, P6, P7, P8, P9) noted that the work required to interact with this menu wouldn't be worth the effort if they had to do it every time they interacted with a data visualization.

Most of our participants who used screen readers (P1, P2, P3, P6, P9) also mentioned that the menu itself was too cumbersome for navigating within and back and forth with the dashboard. Setting options was quite slow, and observing the output of a given option change, such as text verbosity or sonification type, was hard to do with accuracy. Keeping what a previous state was like in memory was hard.

One participant was interested in different ways that this process could become easier, suggesting

> "What if I could just tell it what to change while I'm listening? Like right here [navigating a chart element] what if I could just say "keep it short" or maybe "wait, tell me more."—P6

This suggests that there may be a space to explore non-visual direct manipulation *softerware* strategies.

8.8 Conclusion

In an idealized world, designers do their best to produce useful and accessible interfaces. They're concerned with making software as accessible as possible by default. But no single design is capable of perfection. *Access frictions* between accessible defaults and the needs of real individuals might always be present in software interfaces. To that aim, we hope to contribute knowledge that can inform future designers and developers to not only build accessible artifacts, but build *systems* that enable end users with disabilities to have interactive agency over their software experiences.

Our vision of data visualization *softerware* demands more involvement from research and industry. In order to offer as much value as possible to end-users, we need standards set for accessibility profiles and we need data visualization software, libraries, and applications to respect and be able to contribute to those profiles. We want to encourage researchers to investigate further the needs of people with disabilities, designers to imagine new interfaces and interaction paradigms for end users, and engineers to build robust systems that are capable of not only respecting a user's preferences and customizations, but providing persistence, interoperability, and system-class infrastructure.

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 [37] 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 [98]. 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 [31] or de-centralized agency [13, 54, 73], 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 [106].

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 [84], 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 [115]. 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 *Softerware*, 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 [4] separate from the web? Above it, looking down into it, like shared annotation tools but capable of sharing the manipulation of websites [83]? A space with ambient co-repair, modeled after projects that bring people together [89] or that allow community "fixing" of misinformation [50]? Perhaps feminist thought on the ethics of care can help us [41, 66]? 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] W. Aigner, A. Rind, and S. Hoffmann. Comparative evaluation of an interactive time-series visualization that combines quantitative data with qualitative abstractions. *Computer Graphics Forum*, 31(3pt2):995–1004, June 2012. 8.6.3.2

[2] Obead Alhadreti and Pam Mayhew. Rethinking thinking aloud. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 1–12. ACM, April 2018. doi: 10.1145/3173574.3173618. URL `https://doi.org/10.1145/3173574.3173618`. 6.6.2

[3] E. Bailey. Accessibility preference settings, information architecture, and internalized ableism — ericwbailey.website. `https://ericwbailey.website/published/accessibility-preference-settings-information-architecture-and-interna` 2024. [Online]. 8.5.2

[4] Hakim Bey. *TAZ: The temporary autonomous zone, ontological anarchy, poetic terrorism*. Autonomedia, 2003. 9.3

[5] Brandon Biggs, Christopher Toth, Tony Stockman, James M. Coughlan, and Bruce N. Walker. Evaluation of a non-visual auditory choropleth and travel map viewer. In *Proceedings of the 27th International Conference on Auditory Display (ICAD 2022)*, pages 82–90. International Community for Auditory Display, June 2022. doi: 10.21785/icad2022.027. URL `https://doi.org/10.21785/icad2022.027`. PMCID: PMC10010675. 6.4.1

[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`. 6.3.2

[7] M. Bostock, V. Ogievetsky, and J. Heer. D^{3} 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

[8] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101, January 2006. ISSN 1478-0895. doi: 10.1191/1478088706qp063oa. URL `https://doi.org/10.1191/1478088706qp063oa`. 6.6.3

[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`. 6.3.1

[10] Jennifer L. Burke, Matthew S. Prewett, Ashley A. Gray, Liuquin Yang, Frederick R. B. Stilson, Michael D. Coovert, 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] Matthew Butler, Leona M Holloway, Samuel Reinders, Cagatay Goncu, and Kim Marriott. Technology developments in touch-based accessible graphics: A systematic review of research 2010-2020. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, pages 1–15. ACM, May 2021. doi: 10.1145/3411764.3445207. URL `https://doi.org/10.1145/3411764.3445207`. 6.3.1

[12] F. Cabitza and C. Simone. Malleability in the hands of end-users. In *New Perspectives in End-User Development*, pages 137–163. Springer International Publishing, 2017. 8.4.1.1

[13] 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

[14] C. R. Ceja, C. M. McColeman, C. Xiong, and S. L. Franconeri. Truth or square: Aspect ratio biases recall of position encodings. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1054–1062, February 2021. 8.7.1.2

[15] 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`. 6.3.1

[16] 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`. 6.4

[17] d. akbaba, J. Wilburn, M. T. Nance, and M. Meyer. Manifesto for putting "chartjunk" in the trash 2021! arXiv, 2021. 8.7.4.3

[18] 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`. 6.4

[19] Frank Elavsky and Cindy Xiong Bearfield. Playing telephone with generative mod-

els: "verification disability," "compelled reliance," and accessibility in data visualization. In *2025 IEEE Workshop on Accessible Data Visualization (AccessViz)*, pages 14–24. IEEE, November 2025. doi: 10.1109/accessviz68666.2025.00008. URL `https://doi.org/10.1109/accessviz68666.2025.00008`. 6.3.1

[20] 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`. 6.4.3, 6.8.2, 8.3.1, 8.5.1

[21] 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`. 6.2, 6.3.1, 6.3.2, 6.4, 6.5.1

[22] A. Endert, L. Bradel, and C. North. Beyond control panels: Direct manipulation for visual analytics. *IEEE Computer Graphics and Applications*, 33(4):6–13, July 2013. ISSN 0272-1716. doi: 10.1109/mcg.2013.53. URL `https://doi.org/10.1109/mcg.2013.53`. 6.2

[23] Pavithra Eswaramoorthy, Frank Elavsky, Tania Allard, and gabalafou. Quansight-Labs/bokeh-a11y-audit: v1.0.0, February 2025. URL `https://bokeh-a11y-audit.readthedocs.io/`. 6.4.3

[24] 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, 6.3.2

[25] Ali Mazraeh Farahani, Peyman Adibi, Mohammad Saeed Ehsani, Hans-Peter Hutter, and Alireza Darvishy. Automatic chart understanding: A review. *IEEE Access*, 11:76202–76221, 2023. ISSN 2169-3536. doi: 10.1109/access.2023.3298050. URL `https://doi.org/10.1109/access.2023.3298050`. 6.3.1

[26] Stephen Few. Visual Business Intelligence - Data Visualization and the Blind. `https://www.perceptualedge.com/blog/?p=1756`, 2013. [Accessed 05-04-2025]. 1.1

[27] Danyel Fisher, Igor Popov, Steven Drucker, and m.c. schraefel. Trust me, i'm partially right. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1673–1682. ACM, May 2012. doi: 10.1145/2207676.2208294. URL `https://doi.org/10.1145/2207676.2208294`. 6.2

[28] C. Fleizach and J. P. Bigham. System-class accessibility: The architectural support for making a whole system usable by people with disabilities. *Queue*, 22(5):28–39, October 2024. 8.7.3

[29] 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

[30] 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

[31] David Graeber and Charlie Rose. Never cower to authority, 2006. URL https://www.youtube.com/watch?v=qt5op6wft-8. Accessed [2026]. 9.3

[32] 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

[33] Noor Hammad, Frank Elavsky, Sanika Moharana, Jessie Chen, Seyoung Lee, Patrick Carrington, Dominik Moritz, Jessica Hammer, and Erik Harpstead. Exploring the affordances of game-aware streaming to support blind and low vision viewers: A design probe study. In *The 26th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '24, pages 1–13. ACM, October 2024. doi: 10.1145/3663548.3675665. URL https://doi.org/10.1145/3663548.3675665. 6.2, 6.9, 8.6.2

[34] 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-enviro
2019. [Online]. 8.3.1

[35] G. Harboe and E. M. Huang. Real-world affinity diagramming practices: Bridging the paper-digital gap. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015. doi: 10.1145/2702123.2702561. 8.7

[36] Gunnar Harboe and Elaine M. Huang. Real-world affinity diagramming practices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 95–104. ACM, April 2015. doi: 10.1145/2702123.2702561. URL https://doi.org/10.1145/2702123.2702561. 6.6.3

[37] 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

[38] Gillian R. Hayes. The relationship of action research to human-computer interaction. *ACM Transactions on Computer-Human Interaction*, 18(3):1–20, July 2011. ISSN 1073-0516. doi: 10.1145/1993060.1993065. URL https://doi.org/10.1145/1993060.1993065. 6.2, 6.4

[39] 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. 6.2

[40] 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

[41] 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

[42] Thomas Hermann, Andy Hunt, and John G Neuhoff, editors. *The Sonification Handbook*. Logos Verlag Berlin, Berlin, Germany, December 2011. ISBN: 978-3-8325-2819-5. 6.3.1

[43] L. Hickman and A. Hagerty. Standardised access: the tension between scale and fit. https://www.adalovelaceinstitute.org/blog/standardised-access-tension-scale-fit/, 2021. [Online]. 8.4.1.2

[44] Highsoft. Highcharts - interactive charting library for developers. https://www.highcharts.com/, 2009. [Online]. 8.2

[45] Leona Holloway, Peter Cracknell, Kate Stephens, Melissa Fanshawe, Samuel Reinders, Kim Marriott, and Matthew Butler. Refreshable tactile displays for accessible data visualisation, 2024. URL https://arxiv.org/abs/2401.15836. 6.3.1

[46] Leona M Holloway, Cagatay Goncu, Alon Ilsar, Matthew Butler, and Kim Marriott. Infosonics: Accessible infographics for people who are blind using sonification and voice. In *CHI Conference on Human Factors in Computing Systems*, CHI '22, pages 1–13. ACM, April 2022. doi: 10.1145/3491102.3517465. URL https://doi.org/10.1145/3491102.3517465. 6.3.1

[47] Stacy Hsueh, Beatrice Vincenzi, Akshata Murdeshwar, and Marianela Ciolfi Felice. Cripping 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, 8.3.1

[48] Edwin L. Hutchins, James D. Hollan, and Donald A. Norman. Direct manipulation interfaces. *Human–Computer Interaction*, 1(4):311–338, December 1985. ISSN 0737-0024. doi: 10.1207/s15327051hci0104_2. URL https://doi.org/10.1207/s15327051hci0104_2. 6.2

[49] Hilary Hutchinson, Wendy Mackay, Bo Westerlund, Benjamin B. Bederson, Allison Druin, Catherine Plaisant, Michel Beaudouin-Lafon, Stéphane Conversy, Helen Evans, Heiko Hansen, Nicolas Roussel, and Björn Eiderbäck. Technology probes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 17–24. ACM, April 2003. doi: 10.1145/642611.642616. URL https://doi.org/10.1145/642611.642616. 6.2, 6.9

[50] 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

[51] 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`. 6.3.1, 6.5.2, 8.3.3

[52] 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`. 6.2, 6.3.2, 6.8.2, 8.3.1

[53] 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`. 6.3.1

[54] 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

[55] Gyeongri Kim, Jiho Kim, and Yea-Seul Kim. "explain what a treemap is": Exploratory investigation of strategies for explaining unfamiliar chart to blind and low vision users. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, pages 1–13. ACM, April 2023. doi: 10.1145/3544548.3581139. URL `https://doi.org/10.1145/3544548.3581139`. 6.3.1

[56] Hyeok Kim, Yea-Seul Kim, and Jessica Hullman. Erie: A declarative grammar for data sonification. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, pages 1–19. ACM, May 2024. doi: 10.1145/3613904.3642442. URL `https://doi.org/10.1145/3613904.3642442`. 6.3.1

[57] 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`. 6.3.1

[58] 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`.

6.3.1

[59] N. W. Kim, G. Ataguba, S. C. Joyner, Chuangdian Zhao, and Hyejin Im. Beyond alternative text and tables: Comparative analysis of visualization tools and accessibility methods. *Computer Graphics Forum*, 42(3):323–335, June 2023. ISSN 1467-8659. doi: 10.1111/cgf.14833. URL `https://doi.org/10.1111/cgf.14833`. 6.3.1, 6.3.2, 8.5.1

[60] A. J. Ko et al. The state of the art in end-user software engineering. *ACM Computing Surveys*, 43(3):1–44, April 2011. 8.4.1.1

[61] 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`. 6.2

[62] L. Y. Lo, A. Gupta, K. Shigyo, A. Wu, E. Bertini, and H. Qu. Misinformed by visualization: What do we learn from misinformative visualizations? *Computer Graphics Forum*, 41(3):515–525, June 2022. 8.7.1.2

[63] 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`. 6.3.1

[64] 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`. 6.3.2

[65] M. Maceli and M. E. Atwood. "human crafters" once again: Supporting users as designers in continuous co-design. In *End-User Development*, pages 9–24. Springer Berlin Heidelberg, 2013. 8.3.3, 8.7.1.3

[66] 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

[67] 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`. 6.3.1

[68] 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

[69] Kim Marriott, Bongshin Lee, Matthew Butler, Ed Cutrell, Kirsten Ellis, Cagatay Goncu, Marti Hearst, Kathleen McCoy, and Danielle Albers Szafir. 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. 6.3.1, 8.3.1

[70] Kim Marriott, Matthew Butler, Leona Holloway, William Jolley, Bongshin Lee, Bruce Maguire, and Danielle Albers Szafir. From vision to touch: Bridging visual and tactile principles for accessible data representation. *IEEE Transactions on Visualization and Computer Graphics*, 32(1):659–669, January 2026. ISSN 2160-9306. doi: 10.1109/tvcg.2025.3634254. URL https://doi.org/10.1109/tvcg.2025.3634254. 6.3.1

[71] Andrew M. McNutt. No grammar to rule them all: A survey of json-style dsls for visualization. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–11, 2022. ISSN 2160-9306. doi: 10.1109/tvcg.2022.3209460. URL https://doi.org/10.1109/tvcg.2022.3209460. 6.2

[72] Catherine Mei∗, Josh Pollock∗, Daniel Hajas, Jonathan Zong, and Arvind Satyanarayan. Benthic: Perceptually congruent structures for accessible charts and diagrams. In *Proceedings of the 27th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '25, pages 1–17. ACM, October 2025. doi: 10.1145/3663547.3746342. URL https://doi.org/10.1145/3663547.3746342. 6.3.1, 6.3.2

[73] 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

[74] Peya Mowar, Aaron Steinfeld, and Jeffrey P. Bigham. iTagPDF: Towards finally automating PDF accessibility. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, CHI '26. ACM, 2026. doi: 10.1145/3772318.3790289. URL https://peyajm29.github.io/itagpdf.html. Best Paper Award. 6.3.2

[75] 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

[76] G. Pallapa, S. K. Das, M. Di Francesco, and T. Aura. Adaptive and context-aware privacy preservation exploiting user interactions in smart environments. *Pervasive and Mobile Computing*, 12:232–243, June 2014. 8.3.2

[77] 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. 8.6.2

[78] Lauren Race, Chancey Fleet, Danielle Montour, Lindsay Yazzolino, Marco Salsiccia, Claire Ferrari, Sheri Wells-Jensen, and Amy Hurst. Designing while blind: Nonvisual tools and inclusive workflows for tactile graphic creation. In *The 25th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '23, pages 1–8. ACM, October 2023. doi: 10.1145/3597638.3614546. URL https://doi.org/10.1145/3597638.3614546. 6.4

[79] J.E. Rawling, E.C. Carson, J.W. Attig, D.M. Mickelson, W.N. Mode, M.D. Johnson, and K.M. Syverson. Quaternary geology of wisconsin. Technical report, Wisconsin Geological and Natural History Survey, 2025. URL `https://doi.org/10.54915/xqpw9883`. 6.4.1

[80] 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

[81] Samuel Reinders, Matthew Butler, Ingrid Zukerman, Bongshin Lee, Lizhen Qu, and Kim Marriott. When refreshable tactile displays meet conversational agents: Investigating accessible data presentation and analysis with touch and speech. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):864–874, January 2025. ISSN 2160-9306. doi: 10.1109/tvcg.2024.3456358. URL `https://doi.org/10.1109/tvcg.2024.3456358`. 6.3.1

[82] 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

[83] 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

[84] 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

[85] 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, 6.2, 6.5.2

[86] D. Schepers and F. Elavsky. "accessible data viz". *The Investigative Reporters and Editors Journal*, 4:8–9, December 2024. 8.3.1

[87] Donald Schön and John Bennett. Reflective conversation with materials, April 1996. URL `https://doi.org/10.1145/229868.230044`. 6.8.1

[88] JooYoung Seo, Yilin Xia, Bongshin Lee, Sean Mccurry, and Yu Jun Yam. Maidr: Making statistical visualizations accessible with multimodal data representation. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, pages 1–22. ACM, May 2024. doi: 10.1145/3613904.3642730. URL `https://doi.org/10.1145/3613904.3642730`. 6.3.1

[89] 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

[90] 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. 6.3.2

[91] 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. 6.3.1, 6.3.2

[92] 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. 6.3.2

[93] 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. 6.2, 6.3.2, 6.8.2, 8.3.1

[94] Ben Shneiderman, Christopher Williamson, and Christopher Ahlberg. Dynamic queries. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '92*, CHI '92, pages 669–670. ACM Press, 1992. doi: 10.1145/142750.143082. URL https://doi.org/10.1145/142750.143082. 6.2

[95] V. Sivaraman, F. Elavsky, D. Moritz, and A. Perer. Counterpoint: Orchestrating large-scale custom animated visualizations. In *2024 IEEE Visualization and Visual Analytics (VIS)*, pages 16–20, 2024. 8.3.1

[96] 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. 6.3.1

[97] L. South and M. A. Borkin. Photosensitive accessibility for interactive data visualizations. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–11, 2022. 8.3.1

[98] 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

[99] 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

[100] 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

[101] 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`. 6.3.1, 6.3.2, 6.4

[102] T. Tran, H.-N. Lee, and J. H. Park. Discovering accessible data visualizations for people with adhd. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2024. 8.3.1

[103] W3C. Web content accessibility guidelines (wcag) 2.1, 2018. 8.5.1

[104] W3C. Webaim: Screen reader user survey #10 results. `https://webaim.org/projects/screenreadersurvey10/`, 2024. [Online]. 8.7.2.1

[105] 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`. 6.3.1

[106] 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

[107] Hadley Wickham. A layered grammar of graphics. *Journal of Computational and Graphical Statistics*, 19(1):3–28, January 2010. ISSN 1537-2715. doi: 10.1198/jcgs.2009.07098. URL `https://doi.org/10.1198/jcgs.2009.07098`. 6.2

[108] 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

[109] 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

[110] Leland Wilkinson. The grammar of graphics. In *Handbook of Computational Statistics*, pages 375–414. Springer Berlin Heidelberg, December 2012. ISBN 9783642215506. doi: 10.1007/978-3-642-21551-3_13. URL `https://doi.org/10.1007/978-3-642-21551-3_13`. 6.2

[111] B. L. Wimer, L. South, K. Wu, D. A. Szafir, 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. 8.3.1

[112] Brianna L. Wimer, Ritesh Kanchi, Kaija Frierson, Venkatesh Potluri, Ronald Metoyer, Jennifer Mankoff, Miya Natsuhara, and Matt X. Wang. Nonvisual support for understanding and reasoning about data structures, 2026. URL `https://arxiv.org/abs/2601.19168`. 6.3.2

[113] 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

[114] 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, 8.3.2

[115] 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`. 9.3

[116] R. Wood, J. H. Feng, and J. Lazar. Health data visualization literacy skills of young adults with down syndrome and the barriers to inference-making. *ACM Transactions on Accessible Computing*, 17(1):1–1, March 2024. 8.3.1

[117] 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. 8.3.3, 8.4.1.3, 8.7.4.3

[118] Keke Wu, Emma Petersen, Tahmina Ahmad, David Burlinson, Shea Tanis, and Danielle Albers Szafir. 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. 8.3.1

[119] F. Yanez, C. Conati, A. Ottley, and C. Nobre. The state of the art in user-adaptive visualizations. *Computer Graphics Forum*, December 2024. 8.3.2

[120] 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`. 1.1

[121] Jonathan Zong. Using real names of disabled participant-contributors to practice citational justice in accessibility. In *2025 IEEE Workshop on Accessible Data Visualization (AccessViz)*, pages 30–33. IEEE, November 2025. doi: 10.1109/accessviz68666.2025.00011. URL `https://doi.org/10.1109/accessviz68666.2025.00011`. 6.4

[122] 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`. 6.3.1, 6.3.2, 6.4

[123] Jonathan Zong, Josh Pollock, Dylan Wootton, and Arvind Satyanarayan. Animated vega-lite: Unifying animation with a grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):149–159, January 2023. ISSN 2160-9306. doi: 10.1109/tvcg.2022.3209369. URL `https://doi.org/10.1109/tvcg.2022.3209369`. 6.2

[124] Jonathan Zong, Isabella Pedraza Pineros, Mengzhu (Katie) Chen, Daniel Hajas, and Arvind Satyanarayan. Umwelt: Accessible structured editing of multi-modal data representations. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, pages 1–20. ACM, May 2024. doi: 10.1145/3613904.3641996. URL `https://doi.org/10.1145/3613904.3641996`. 6.3.2, 6.8.4