

Chart Reader: Accessible Visualization Experiences Designed with Screen Reader Users

John Thompson*
johnthompson@microsoft.com
Microsoft Research
Redmond, Washington, USA

Jesse Martinez*
jessejm@cs.washington.edu
University of Washington
Seattle, Washington, USA

Alper Sarikaya
alper.sarikaya@microsoft.com
Microsoft Corporation
Redmond, Washington, USA

Edward Cutrell
cutrell@microsoft.com
Microsoft Research
Redmond, Washington, USA

Bongshin Lee
bongshin@microsoft.com
Microsoft Research
Redmond, Washington, USA

ABSTRACT

Even though screen readers are a core accessibility tool for blind and low vision individuals (BLVIs), most visualizations are incompatible with screen readers. To improve accessible visualization experiences, we partnered with 10 BLV screen reader users (SRUs) in an iterative co-design study to design and develop accessible visualization experiences that afford SRUs the autonomy to interactively read and understand visualizations and their underlying data. During the five-month study, we explored accessible visualization prototypes with our design partners for three one-hour sessions. Our results provide feedback on the synthesized design concepts we explored, why (or why not) they aid comprehension and exploration for SRUs, and how differing design concepts can fit into cohesive accessible visualization experiences. We contribute both Chart Reader, a web-based accessibility engine resulting from our design iterations, and our distilled study findings—organized by design dimensions—in the creation of comprehensive accessible visualization experiences.

CCS CONCEPTS

• **Human-centered computing** → **Visualization design and evaluation methods**; **Accessibility systems and tools**.

KEYWORDS

accessibility, data visualization, blind and low vision, screen readers, iterative co-design, accessible visualization experiences, accessibility engine

ACM Reference Format:

John Thompson, Jesse Martinez, Alper Sarikaya, Edward Cutrell, and Bongshin Lee. 2023. Chart Reader: Accessible Visualization Experiences Designed with Screen Reader Users. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*, April 23–28, 2023, Hamburg, Germany.

*Both authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CHI '23, April 23–28, 2023, Hamburg, Germany

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-9421-5/23/04...\$15.00

<https://doi.org/10.1145/3544548.3581186>

Germany. ACM, New York, NY, USA, 18 pages. <https://doi.org/10.1145/3544548.3581186>

1 INTRODUCTION

Data visualization enables people to efficiently explore data and effectively communicate insights. However, due to its inherent reliance on human visual capabilities, data visualization is not readily accessible to blind or low vision individuals (BLVIs). Screen readers are a core assistive technology tool for BLVIs, which announces digital content as synthesized speech. They, however, are optimized for reading structured document content, which is at odds with spatial, temporal, and non-linear forms of multimedia (e.g., images, videos, maps, charts). Most data visualizations, even on mainstream websites, are incompatible with screen readers [27].

BLVIs who use a screen reader experience one of the following circumstances when they encounter a web-based visualization: (1) nothing (undiscovered by the screen reader), (2) a textual description (experiences vary, e.g., from “object” to “an image of a bar chart,” to rich descriptions of the visualization including meaningful insights, such as overall trends), and (3) interactive or explorable descriptions of the visualization and underlying data. Sometimes, a data table (or downloadable data file) is provided in lieu of or in addition to a visualization.

Obviously, undiscoverable charts or nonsensical descriptions of them are the worst-case scenarios for screen reader users (SRUs). As a baseline, accessibility guidelines recommend visualization authors provide textual or tabular representations of visualizations as an alternative for SRUs. Textual descriptions of a visualization are most useful when they “report statistical concepts and relations” or “identify perceptual and cognitive phenomena” [18]. At the other end of the spectrum, data tables provide an unrestricted opportunity to explore the data. The underlying data table can be read, analyzed, sonified, or printed as a tactile chart. However, this approach requires technical expertise and puts an undue burden on BLVIs, as they must invest time to learn and implement data exploration strategies. Furthermore, providing tabular representations is a lossy exchange, as raw data is devoid of the design decisions and communicative intent of the author, giving up the benefits of data visualization. Combining the advantages of tabular and textual representations, interactive accessible visualizations enable SRUs to explore descriptions of the chart and its underlying data, their derived values or structures, and author-identified insights.

Recently, visualization research and practice has made strides in several avenues toward interactive visualization experiences that integrate with screen readers. Web-based component libraries [10, 33] allow SRUs to explore the data and rich alternative descriptions of a chart through keyboard navigation. Building on these practices, Zong *et al.* [38] prototyped visualizations with varying structure and keyboard navigation experiences through a co-design study, identifying the importance of boundaries in exploration. Beyond textual descriptions, non-speech audio of the data (sonification) [10, 29, 30] can communicate quantitative data values by mapping them to tonal pitch or illustrate matching points of interest (line intersections) with earcon sounds. In our work, we strive to synthesize these previously disparate works together to understand how to balance features with usable designs of visualizations for SRUs: our goal is to provide an accessibility engine that synthesizes best practices for alt-text description, a combination of hierarchical and flexible navigation among chart components, and non-speech sonification of data characteristics without requiring a specific screen reader, extension plugin, or additional hardware. With this engine, we can explore the holistic design space of accessible visualization representations.

To synthesize, situate, and expand upon the combination of multiple accessible visualization techniques, we conducted an iterative co-design study with 10 BLV screen reader users over a continuous period of five months. Our design partners were very passionate about our co-design study, and they represented a significant diversity in perspectives and expertise, as well as in demographics. This greatly benefitted our design collaborations, enabling us to conduct co-design at a very high level of comprehensiveness. We had three one-on-one sessions (once every 4–6 weeks) with each design partner, during which they interacted with accessible visualizations. We presented an initial version of our accessible visualization prototype in the first session, and refined the prototype between sessions based on feedback and ideas gathered from previous sessions. Our accessible visualization began with basic line charts and extended to cover multi-series line charts and stacked bar charts. We used visualization prototypes as probes, not only to ground the feasibility of ideas but also to contextualize tasks, interactions, and interfaces with real data. Our iterative co-design study yielded a multitude of accessible visualization design ideas, feedback on the implementation of many of those ideas, and accessible visualization designs that can transfer to similar chart types.

The main contributions of this work are twofold. We present the design and implementation of Chart Reader, a web-based accessibility engine, which enables rendering of accessible visualizations for BVLIs to read and better understand the visualizations and their underlying data (Section 4). Chart Reader extends the state-of-the-art work [29, 30, 38], incorporating non-speech audio and alternative ways to navigate chart components. We also contribute the understanding of helpful (and unhelpful) designs from our iterative co-design study on accessible, usable, and effective screen reader interfaces for experiencing data visualizations (Section 5). We present these insights as lessons that we learned and can be applied when creating other accessible visualization experiences.

2 BACKGROUND AND RELATED WORK

2.1 The Web Accessibility Initiative (WAI)

In this paper, we focus on web-based data visualization. Visualizations built on the web can take advantage of accessibility interfaces built into browsers that adhere to web and web-adjacent specifications. Specifically for accessibility, the WAI-ARIA specification [34] uses attribute tags intertwined in HTML’s DOM (document object model) to generate a generalized accessibility tree that screen readers can consume without the need for specialized plugins. By using the right combination of labels, roles, and associated details, linearized navigation of a complex document is possible. While guidance is given by WAI-ARIA for complex images [35], its suggestions are limited to textual descriptions of visualizations (alternative text, often shortened to “alt text”) and makes no note of interaction or piecemeal consumption paradigms. In addition to screen reader support, the modern web stack also has support for generating audio programmatically via WebAudio [37], allowing authors to synthesize tones, chords, and melodies dynamically generated from document content.

Unfortunately for visualization viewers, the “correct” usage of these accessibility and audio standards for visualization consumption is underdefined. While many visualizations do not support any sort of interaction or interrogation, there are also many examples where misuse of ARIA tags lead to poor consumption experiences [7]. Efforts such as the early WAI-ARIA Graphics specification [36] and directly modifying the Accessibility Object Model [2] strive to standardize and enrich consumption and authoring paradigms, but are still in their infancy.

2.2 Accessible Visualization Experiences

A large part of the consumption and interaction paradigms associated with data visualization are intertwined with the input and output modalities of mouse, keyboard, touch, and graphical display devices [14]. As a result, generalized knowledge about how to best support both authors and consumers of data visualizations using other modalities as well as assistive technology has lagged behind these traditional modalities [20]. However, recent work has re-energized interest in this area in visualization authoring for accessibility, particularly in pushing forward consumer interaction paradigms with visualizations [4, 12, 14, 38].

The proliferation of web-based visualizations has pushed the focus for consumable, accessible visualization experiences towards web-based accessibility standards. Many commercial applications are also constructed based on these web standards, such as Audio Graphs [1], Highcharts [10], Microsoft Power BI [21], SAS Graphics Accelerator [25], and Tableau [32]. These tools provide varying support for consumption experiences, which often depend on the author to provide relevant alt text for individual visualizations. Between these applications, there is varying support for interacting with chart elements such as axes, series, data points, and high-level trends. Visualization grammar-based tools such as Vega-Lite have also added support for ARIA labels [26], but its intended usage and impact to navigation is underdefined.

There has been excitement in recent research regarding how to best support and generalize accessible visualization and data experiences. Early studies such as those constructed by Brewster [3]

point at the need to consider multiple modalities when constructing accessible experiences, while recent work by Elavsky *et al.* [5] provide prompts for visualization authors to think beyond the minimum accessibility specification towards usable experiences. Kim *et al.* [14] describe the breadth of the design space from existing web-based visualizations and highlight areas for further study, including interaction and consumption representations. Chundury *et al.* [4] use semi-structured interview with blind orientation and mobility instructors to understand how blind individuals perceive spatial concepts, and discuss implications for visualization design for blind individuals. We adapt some of their high-level insights to our work, such as minimizing sensory overload, not creating new barriers between BLVIs and others, and supporting interaction and navigation as a critical piece for non-visual sensemaking.

Studies specific to particular modalities such as speech with VoxLens [29] or to specific visualization types such as line charts [23] and interactive maps [9] validate how core design choices affect the data consumption experience. All of these studies point at the need to mix modalities to support the wide range of visualization consumers. However, the need to textually *describe* trends, context, and visualization construction is pervasive. Jung *et al.* [13] explore the level of textual support individuals need and find superfluous in describing a data visualization, while Lundgard *et al.* [18] categorize “levels” of descriptions from descriptions of visual encodings to descriptive statistics, and to complex trends and high-level insights. While alt text in and of itself does not replace consumers’ agency of free-form visualization navigation, these works provide a strong foundation for visualization authors to ground their component descriptions upon. For auditory cues, Holloway *et al.* [11] introduce *Infosonics*, which mixes interaction with multiple audio tracks to interpret trends in an infographic. Siu *et al.* [30] gear more toward audio narratives, which provide guidelines for duration, complexity, contextual clues, and mixing with textual descriptions. Sharif *et al.* [28] also discuss SRUs’ preferences for non-speech audio cues, finding that there is a preference towards non-continuous square waveforms—a data sonification strategy that we adopt in our prototype. On the whole, these works hint at the need to support individual differences on behalf of visualization viewers and multiple avenues to access relevant segments of data.

In moving towards complete accessible parity for interacting with data visualizations, Sharif *et al.* [27] identify the inherent limitation of screen readers in “linearizing” interaction patterns. They point to the need to hierarchically organize interactive elements in a discoverable and consumable manner. Fan *et al.* [6] find the prevailing issues with accessibility support for popular web-based visualizations, highlighting that consumption experiences are poor to non-existent and that simple “adherence to the spec” strategies are not sufficient for SRUs. They advocate for multiple representations (for viewer diversity, task, and goal), combining screen reader and other auditory representations, and embracing other web-based modalities to create a complete data experience. Most similar to our work, Zong *et al.* [38] describe a manner of hierarchically organizing the interaction of a screen reader into chart components such as axes, legend, data points, and annotations. Using the accessible primitives of tables, trees, and lists, they lean toward trees to flexibly arrange screen reader and iterative interaction with the

visualization. While this work concentrated on hierarchical organization and textual descriptions of chart components, we extend this state-of-the-art work to consider alternate modalities such as sonification and alternative ways to navigate chart components.

2.3 Co-Design Methods

Co-design and inclusive design are powerful strategies for ensuring that user goals, intents, and interaction preferences are incorporated into a user interface [15, 24]. Through the inclusion of users in the design process, participants help realize potential beyond what designers may have originally envisioned for the interface [15]. Specifically for accessible co-design, Mankoff *et al.* [19] clarify that assistive technology researchers can identify technical and social limitations of their work by understanding disability studies as a form of critical inquiry. While the use of co-design is attractive to researchers and designers, Lundgard *et al.* [17] raise socio-technical concerns of designing accessible data visualizations, specifically highlighting the need to reduce barriers to entry by following existing accessibility standards, clearly communicating intents, and fairly including potential users in the design process.

Similar to our study, Zong *et al.* [38] used an iterative co-design method to understand consumption limitations with existing practices. The design iteration in that study helped to identify issues with describing semantic content represented by components in the visualization, while also considering competing goals of expressiveness, verbosity, context, and flexible navigation. The authors then evaluated three of their prototypes with 13 users not involved in the co-design process to validate whether the design choices they made extended to a diverse set of users. In contrast, our study incorporates 10 participants with diverse expertises and backgrounds in parallel co-design over three iterations, where each iteration incorporated design responses to summative feedback (see Section 3). While more complex in nature, we believe such a process provides better understanding of individual differences, and helps to point design toward more flexible interfaces that promote agency on behalf of the visualization consumer (see discussion in Section 6.3).

2.4 Initial Chart Reader Design

To guide the overall design of Chart Reader, we developed five core design dimensions derived from the set proposed by Zong *et al.* [38], and supplemented by other prior work [4, 11, 13, 18, 27, 29, 30] and some specific design needs of our tool. As shown in Table 1, our scope of design dimensions includes Zong *et al.*’s dimensions of *structure*, *navigation*, and *description*, as well as *non-speech audio* (e.g., sonification, earcons) and *focus* (e.g., filtering, highlighting, selection). As a prompt for our initial co-design session with participants, we developed an initial “v1” accessible visualization prototype (rendered using Chart Reader) based on these dimensions and the findings of prior work. Our intent was to synthesize features of prior work in a single tool without requiring the consumer to use a specific screen reader, extension plugin, or additional hardware. The initial prototype only supported augmenting a continuous, single-series line chart (the chart type most frequently supported in prior work), and included the following:

Table 1: An overview of the design dimensions we considered when architecting a comprehensible screen reader experience.

Dimension	Description & Example
Structure	How chart elements are organized in a format to be navigated with a screen reader [27, 38]. George enters the first-level of the control hierarchy, where each component affords different access modalities to the chart: “ <i>data insights, x axis, y axis, data points</i> ” (Section 4.3)
Navigation	Methodology to freely traverse the accessible structure of a visualization [38]. Emma is navigating along the “ <i>x axis bin</i> ” for “1999.” She moves to the <i>Right</i> , hearing “2000, 2001, 2002, ...” as she moves along the bins (Section 4.4)
Description	Textual representations of all chart elements [4, 13, 18, 30, 38]. George reads the first insight: “ <i>Second Largest Peak, Winter 2021...</i> ” George cuts off the announcement early as he is interested in more recent events, and navigates to the next insight (Section 4.3)
Non-speech audio	Auditory stimuli such as tones or music that convey chart elements, state, or data characteristics [4, 11, 29]. When Murphy presses <i>Right Arrow</i> repeatedly along the x axis, the chart emits a “ <i>Bonk</i> ” sound that indicates a boundary. From this notification, Murphy knows she is at the end of the “ <i>x axis</i> ” (Section 4.5)
Focus	Mechanism to modify or remove chart elements from the navigable structure to improve clarity [4]. Emma uses series filter functionality to select two out of four series and sonify the data points of just those two series concurrently to identify correlation (Section 4.4)

- **Hierarchical chart structure:** chart elements are hierarchically organized based on their arrangement: annotations, axes, and data points,
- **Keyboard navigation:** a user can use *Enter* and *Esc* to move up and down the hierarchy, and use *Up Arrow* and *Right Arrow* or *Down Arrow* and *Left Arrow* to move between items,
- **Annotations:** a user can interact with multiple text elements that provide different insights into the features contained within and context around the data visualization,
- **Element descriptions:** announce element type, associated description, and any data bindings (i.e., category and value for a line chart vertex), and
- **Sonification playthrough:** when a user presses Shift+Enter, Chart Reader plays a sonification of the series, with a tone for each available value encoded in the tone’s pitch.

3 OUR METHOD: PARALLEL CO-DESIGN

We designed and implemented Chart Reader, a web-based accessibility engine that renders accessible visualization experiences, with 10 BLV design partners over five months. In this section, we describe how we adopted a co-design method to iteratively develop accessible visualization experiences.

We conducted our co-design sessions as a series of one-on-one sessions with each partner, which we call a parallel co-design study. Having individual sessions instead of group sessions enabled us to directly work with each design partner to gather their insights and comments, without them being influenced by other partners’ perspectives. We found this helped to identify their personal design preferences. Furthermore, this structure enabled us to choose which portions of the design to focus on with each participant during each session, recognizing that different participants had different areas of expertise within this design domain.

We conducted all sessions remotely via the Microsoft Teams videoconferencing software, and recorded the video call sessions

using its recording feature. As our accessible visualization prototypes were web-based, this virtual format allowed us to integrate live exploration of the prototypes into our general design conversations, while also permitting our partners to use their own software and hardware setups (including personalized screen reader settings, Braille displays, and a high-contrast and magnified visual display).

3.1 Co-Design Partners

We recruited 10 BLV design partners (7 males, 3 females) from Microsoft by using an internal mailing list. To be eligible to participate in the study, they identified themselves as blind or low vision and frequently used a screen reader to navigate the web. Our design partners’ ages were spread across five age groups: 25–34 (3), 35–44 (2), 45–54 (2), 55–64 (2), and 65+ (1). Beyond diversity in demographics, they represented a significant diversity in perspectives and expertise as summarized below, which benefited our design collaborations and enabled us to conduct co-design at a high level of comprehensiveness.

- **Assistive Technology Setups.** Our design partners use various screen readers—NVDA (9), JAWS (5), and Narrator (6)—with many different configurations. They ran them at speeds ranging from less than 1x speed up to about 4x (this is our estimation). Several partners also use Braille displays as a complement to their screen reader, and one partner used a high-contrast, magnified screen.
- **Expertise on Assistive Technology.** Several partners had extensive (e.g., 18 years) experience in the design or development of assistive technologies including screen readers. One design partner helped to design or develop accessible applications as an advisor and reviewed many accessibility specifications, and another partner taught assistive technology for over 15 years.
- **Screen Reader Experience.** Our design partners had a wide range of experience and familiarity with screen readers. Some utilized advanced hotkeys and informed our research

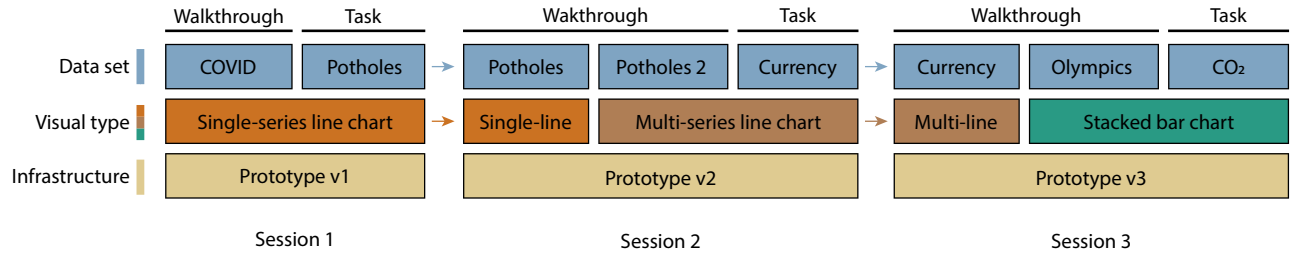


Figure 1: Accessible visualization prototypes, visualizations types, and datasets used in each session.

team of the inner workings of their screen readers, while a few had to wrestle with their screen readers a bit more to accomplish their intended actions.

- *Data and Chart Familiarity.* Some design partners were well-versed in data exploration, regularly working with raw data and data tables, both professionally and recreationally. Others had either basic knowledge of some chart types with infrequent use, or little to no knowledge of charts and generally avoided engaging with them.
- *Knowledge of Other Tools.* Some partners were well-versed in chart tools, including HighCharts [10], Audio Graph [1], Excel [22], SAS Graphics Accelerator [25], and Desmos Graphing Calculator [31].
- *Diversity of Visual Impairments.* Our design partners had a range of visual impairments, which naturally affected their experience with our tool and, as some partners emphasized, their familiarity with chart layouts and visual metaphors used in data visualizations. Three partners were born blind, three lost vision in childhood (before the age of 18), three lost vision in adulthood, and one partner had low vision.

In addition to their diverse disability identities supporting comprehensive co-design that generalizes for people with disabilities [19], their diverse expertise in other areas enabled us to deeply engage in different topics with each partner. For instance, with a partner with a background in assistive technology development, we were able to brainstorm implementation details of a particularly challenging feature (which was not a topic we could discuss with all our partners). Other unique topics we explored with specific partners include: how our prototype can scale to support large-scale data (discussed with two partners who professionally manage large datasets), and how it could be integrated into existing research publication accessibility practices (discussed with a partner who professionally engages with research papers, many of which contain inaccessible charts).

Overall, our design partners enabled us to consider many more design perspectives than we initially anticipated, which we were able to incorporate into our final design to produce a prototype that is not just accessible, but also well-rounded.

3.2 Iterative Co-Design Sessions and Data Analysis

We conducted our co-design process iteratively, with three distinct sessions for each design partner to participate in. For each session,

all design partners worked with the same version of the accessible visualization prototype employing the same data visualizations and underlying data (Figure 1). After all available design partners completed a session, we coded the feedback and design insights from our design partners. We then updated the Chart Reader prototype with new features and modifications based on the feedback and insights from the session, and used this updated prototype in the following session. Each version of the prototype also introduced a new chart type: the first prototype only supported single-series line charts, the second prototype added support for multi-series line charts, and the third prototype added support for stacked bar charts. As mentioned above, the initial chart type was chosen to best incorporate prior work, and subsequent chart types were selected to extend the previously introduced type (i.e., adding multi-series support to a line chart, and adding categorical X-axis support to a multi-series chart). In Section 5, we report the set of accessible visualization design ideas we tried in each session and how they evolved throughout the three sessions.

Procedure. Each session consisted of three segments—(1) pre-session information gathering, (2) feedback and idea generation from experiencing prototypes, and (3) debrief—and started with our partners sharing their computer’s screen and audio. Even though the main structures of the three sessions were very similar, the first session was different from the other two sessions in two aspects. The first session started with questions to gather general background information (e.g., gender, age group, screen reader experience, technical expertise) and the description of our co-design study logistics (format, cadence, compensation, etc.), whereas the other two sessions started with questions regarding their experiences with data visualization, if any, between sessions. In addition, in the first session, we presented an initial version of the accessible visualization prototype we prepared; and thus it was new to our design partners. In the other two sessions, design partners first revisited one of the two visualizations from the previous session using the updated prototype, providing us with feedback on the changes made between sessions (Figure 1). The full set of visualizations used in our study can be found in our supplemental materials. In all sessions, our partners experienced two instances of the same visualization type—we used the first one to walk them through the prototype and they used the second one on their own.

When using the visualization prototype on their own, we provided design partners with two questions (Table 2) to encourage their engagement with the prototype. Because our intentions were to encourage design partners to explore the visualization and to

Table 2: Questions provided to design partners when they were using the visualization prototype on their own for each session.

Session #	Questions
Session 1	Q1. Which month had the maximum number of reported potholes? Q2. Is there a seasonal trend when potholes are reported? If so, please describe the seasonal trend.
Session 2	Q1. What is the maximum value on this chart? What series has the highest value, and in what month/year? Q2. At what points did the New Zealand Dollar have a higher value than the Brazilian Rial?
Session 3	Q1. Which country has the most transportation emissions? Q2. Which country has the most equal distribution within the sectors?

understand how they interacted with the prototype to complete the same tasks, we asked them to describe their strategy or intention, if possible, and allowed them to ask questions to us.

Of 10 partners, seven participated in all three sessions, two participated in only Sessions 1 & 3, and one participated in only Sessions 2 & 3. For the three participants who participated in only two sessions, the session after the one they missed was modified into a “combined” session, where they were introduced to features from both the previous (missed) session and the current session. Even though we made previous versions of our accessible visualization prototype available throughout the five-months study, only two design partners very briefly tried earlier versions between sessions.

Data Analysis and Prototype Refinement. To identify relevant design takeaways from each session, we transcribed the recording of each session and coded the transcripts. For our coding, we used a deductive coding process with the following five codes: 1) positive interactions, 2) negative interactions, 3) unusual experiences and perspectives, 4) desired or requested features, and 5) bugs. With this high-level coding, we then subsequently designed features and modifications for the next prototype that would address pain points and bugs (Codes 2 & 5) and incorporate desired features in modes that had been well-received (Codes 1 & 4). Code 3, “unusual experiences and perspectives,” most frequently highlighted partners’ unique backgrounds that informed their recommendations, such as when a design partner with experience developing assistive technology recommended a means of handling a particular bug, or when a partner using a Braille display highlighted inconsistencies specific to the relationship between their various assistive technologies.

4 CHART READER: RENDERING ACCESSIBLE VISUALIZATION EXPERIENCES

In this section, we first present a high-level overview of the Chart Reader engine and then describe three usage scenarios of accessible visualizations rendered using Chart Reader.

4.1 Chart Reader Overview

4.1.1 Chart Reader Implementation. We implemented Chart Reader as an accessibility engine that renders an SVG chart to a web page using three inputs called from JavaScript code. The engine instantiates event handlers to react to user keystrokes that are translated into actions (e.g., navigate, sonify, filter). The inputs to the accessibility engine are: 1) a data file (in CSV format), 2) an insights list (in JSON format), and 3) a chart configuration (in JSON format). The data and set of insights describe the content of the accessible chart,

while the chart configuration declares how the data is presented to the screen reader.

Data. In the data file, Chart Reader supports the following data fields: number, string, datetime, date, and time. The engine expects data to be complete and tidy: it currently does not support missing values.

Insights. The insights JSON structure builds on the d3-annotation library [16]. Four fields are necessary for Chart Reader to provide a screen reader experience: title, label, target, and type. Title and label are textual descriptions announced by the screen reader. Target is the data targeted by the insight, specified by the axis and values under selection (e.g., {target: {axis: "x", values: ["2020-03-01", "2020-04-10"], series: ["Seattle"]}}). Values can either be a range in the case of number or temporal data types, or a list of values in the case of a string data type. Series is a list of strings in the case of multi-series data. The insight type describes how the insight should be grouped (e.g., "Summary", "Trends"). Insight types are strings that can be added ad-hoc by simply including new types in the file: Chart Reader will group any insights together by type.

Configuration. The chart configuration JSON object is a declarative specification of the chart. While limited in scope, the configuration object is similar to other declarative chart specifications [26]. The configuration includes a description object to include the chart’s title and caption that Chart Reader announces in a high-level summary description of the chart. The configuration also requires specifications for each data encoding, depending on chart type. For example, x, y, and z are required for a stacked bar chart, while only x and y are required for a single-series line chart. The encoding declares the following fields:

- encode specifies the channel to encode (e.g., x, y, z);
- name declares the column name to use from the data table;
- type declares the data type of the column (e.g., number, string); label_axis, label_tooltip, and label_group are open-ended strings that will be announced as labels to values in the chart (e.g., {label_axis: "Number of pothole reports"});
- interval describes the temporal or numerical interval between axis bins (e.g., {interval: "Month"}); aggregate declares the function to aggregate data within axis bins (mean, max, min, sum, count, consecutive_datetime); and
- period describes the frequency of data values (Second, Minute, Hour, Day, Week, Month, Year) for temporal data types.

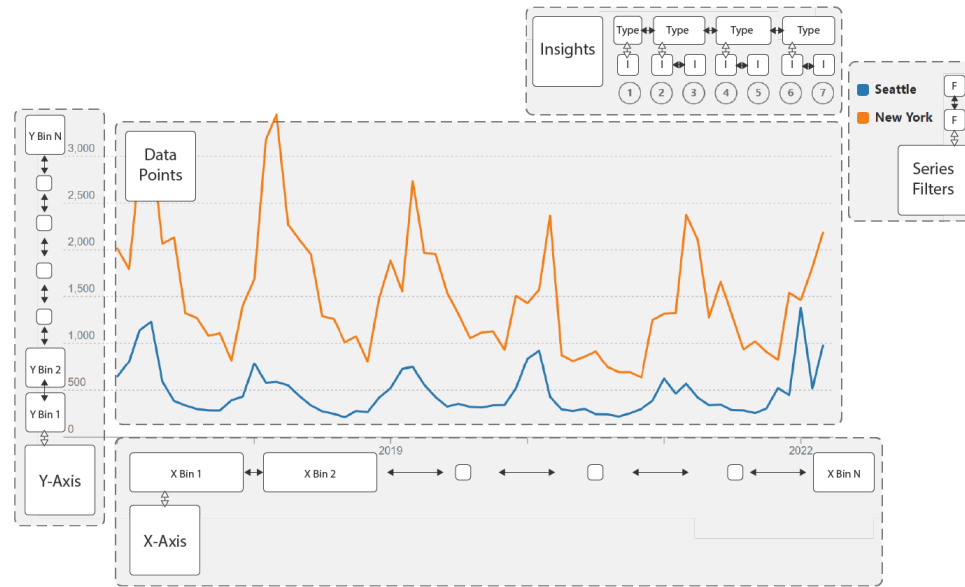


Figure 2: Five regions in charts rendered using Chart Reader to support accessible visualization experience. Data Insights, with sub-regions for each insight type and insight, further subdivided for each individual insight. X-Axis, with sub-regions for each bin along the axis. Y-Axis, with sub-regions for each bin along the axis. Data Points region. Filters, with sub-regions for each individual series.

4.1.2 Accessible Visualization Experience Structure. Accessible visualizations rendered by Chart Reader have the same general structure, depicted using an annotated screenshot and a flow diagram in Figures 2 and 3, respectively. When embedded in the page, each chart has a high-level summary description that also informs users of how to begin interaction. The level below that in the hierarchy contains five main branches: Data Insights, X-Axis, Y-Axis, Data Points, and Filters.

Data Insights. The top level of the Data Insights region provides a description of how many Insights are included in the chart, and how many insights there are of each type. The next level down in the hierarchy contains a group for each insight type, and the level below each group contains the individual insights. Users can navigate from the individual insights directly to the associated portion of the Data Points.

X- and Y-Axes. The top level of each Axis provides a description of the axis. The next level down contains bins for each axis, defined by the interval described in the chart’s configuration. On a numerical X-axis, these bins include the average value within the bin for each series; on a categorical X-axis, these bins include the total values within the bin for each series. On the Y-axis, these bins include the percentage of the data of each series that falls within this interval. Users can navigate from the bins on each axis directly to the associated portion of the Data Points.

Data Points. The Data Points region navigates the users directly to the individual data points within the series. Users can navigate through the points of a selected series (ordered by X-value), or hop from one series to another at the currently focused X-value.

Compare Series. The Compare Series region navigates the users directly to all series of a selected data point (ordered by X-value). Users can sonify all series of the selection as spatialized tones. This region is available for multi-series chart types.

Filters. The Filters region contains a checkbox list of all series available in the data. Each checkbox can be toggled on or off to determine if that series is present in the rest of the chart. This region is available for multi-series chart types.

Added Navigation. In addition to the basic structural navigation described above and in Figure 3, Chart Reader supports several other navigational commands. By pressing *Home* or *End*, users can navigate to the first or final element of a group, respectively. Similarly, *Page Up* and *Page Down* allow the users to hop five elements forward and backward. Finally, Chart Reader incorporates shortcut keys to allow for easier navigation and reorientation: each top-level region of the chart can be jumped to with a single key: *I* for Insights, *X* for the X-Axis, *Y* for the Y-Axis, *D* for the Data Points, and *F* for the filters.

Added Sonification. In addition to the original prototype’s incorporation of full-series sonification playthrough, our final Chart Reader prototype enables two additional types of sonification: pointwise sonification and spatial comparison. Pointwise sonification allows users to manually sonify a single point as they navigate through a chart. Spatial comparison allows users to sonify the value of each series at the currently focused x-value through a set of discrete, sequentially played tones that are spatialized using stereo audio. Both sonification styles are described in more detail in Section 4.2 and in our supplemental materials.

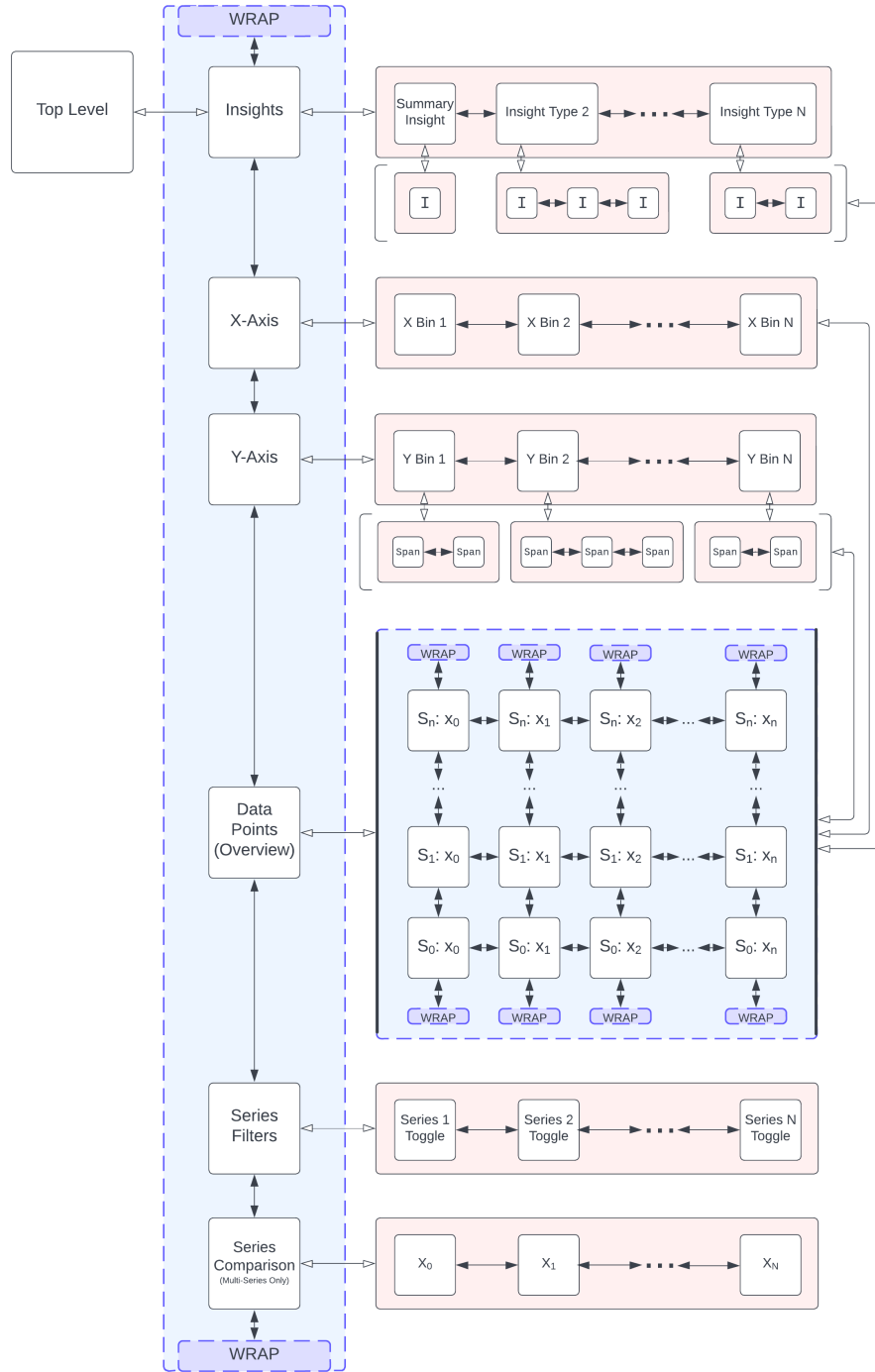







Figure 3: Flow Diagram depicting the structure and navigational flow of an accessible visualization experience rendered using Chart Reader. Starting at the Top Level node, a user first selects a high-level Region to explore, and can progressively focus further in on this Region (denoted by the white-tipped arrows). Ultimately, all Regions aside from the Filters and Series Comparison focus down to the individual data points. At any point, a user can also return to the higher-level node they were previously on.

Table 3: A sample of the features we built into the prototypes, along with the dimensions they are intended to support. We expand on several of these features in Section 5.

Feature description	Prototypes			Design dimensions
	v1	v2	v3	
Structure of chart				
Hierarchical tree structure (parent child sibling elements)				Structure
Top level elements: insights, x axis, y axis, data points				Structure
Top level (added) elements: compare data, filter				Structure
Insights are parents of targeted data				Structure
Y bins are parents of flattened spans				Structure
Y bins are parents of multiple spans				Structure
Y bins are parents of co-occurring series spans				Structure
X bins are parents of binned data				Structure
Navigate hierarchy structure with keystrokes				
Enter (down level), ESC (up), Left (left sibling), Right (right)				Structure, Navigation
Up (next sibling), Down (previous sibling)				Structure, Navigation
Up (next series), Down (previous series)				Structure, Navigation
Home (first sibling), End (last sibling)				Navigation
Page Up (left 5 siblings), Page Down (right 5 siblings)				Navigation
Hot Keys to Top Level Components (e.g., I to Insights, X to X-axis)				Navigation
Navigation patterns				
Looping at the start end of list				Structure, Navigation
Looping at series navigation				Navigation
Describe insights				
Similar to annotations				Description
Framed by template structure				Description
Describe data points				
Order by x-value, y-value, series-value				Description
Order by y-value, x-value, series-value				Description
Order series first when navigating by series				Navigation, Description
Describe the chart				
Signpost at start, describe data and chart elements				Structure, Description
Signpost at exit				Structure, Description
Include summary insight				Description
Sonification and Earcons				
Play element data unless interrupted (Shift & Enter)				Structure, Non-Speech Audio
Interweave data value announcements with sonification				Structure, Navigation, Non-Speech Audio
Announce last data value on pause				Structure, Navigation, Non-Speech Audio
Spatial sonification of series				Structure, Navigation, Non-Speech Audio
Navigate while sonifying (Shift + <Navigation Key>)				Navigation, Non-Speech Audio
Play "bonk" sound at boundary				Navigation, Non-Speech Audio
Play "drip" sound for invalid sonify				Navigation, Non-Speech Audio
Document mode				Navigation, Focus
Filter series				Structure, Focus

4.2 Introduction of Accessible Visualization Experiences

To provide accessible experiences for multiple chart types, we built a rich set of features into Chart Reader. Table 3 shows a sample of these features, along with the dimensions they are intended to

support. In the remainder of this section, we present these features through three example scenarios. (For alternative presentations of the features of Chart Reader, please refer to our supplementary material for a manual-style listing and description of the features in the final design, or for interacting with charts rendered with our design prototype first-hand.) In each scenario, a different SRU

persona uses our design to interact with one of the three example charts from our study:

- (1) **“COVID single-line”**: a single-series line chart displaying COVID-19 reported cases in the United States (described in Section 4.3, shown in Figure 4)
- (2) **“Currency multi-line”**: a multi-series line chart displaying global currency values (described in Section 4.4, shown in Figure 5)
- (3) **“Olympic stacked-bar”**: a stacked bar chart displaying Olympic medals won by countries in the Summer 2020 Games (described in Section 4.5, shown in Figure 6)

We derived our three SRU personas from the predominant strategies that our design partners employed to find information from visualizations and understand data. We observed (and in many cases, our design partners called out) three strategies, introduced here as personas: (1) **George** comprehends best from rich textual descriptions, (2) **Emma** interrogates chart structure and underlying data to build-up a higher-order understanding of the data, and (3) **Murphy** best perceives data through non-speech audio. We note that these strategies are not mutually exclusive to these personas, or the design partners they are based upon. These predominant strategies were often used when the persona started their process, either to glean an overview of the chart or answer a specific question. Secondary strategies were used to verify or find undiscovered information yielded by their primary strategy. The charts described have the same interface and interactive mechanics, and they are not customized for each persona. Our design is meant to generalize, yet afford each persona their preferred strategy to explore and comprehend the chart.

4.3 Experience 1: Single-Series Line Chart

George learns best from rich textual information that describes the salient aspects of a chart. To understand a chart about COVID-19 reported cases in the United States as a 7-day rolling average (Figure 4), he first reads the caption for the chart: “Header 2. COVID-19 Cases in the United States. This chart communicates reported COVID-19 cases as a 7-day rolling average.” George then reads the chart’s high-level summary description: “The visualization is a line chart with 1 series. The chart has 1 x-axis displaying days from the pandemic start. The date ranges from January 21, 2020 to April 30, 2022. The chart has 1 y-axis displaying reported cases as a 7-day rolling average. The cases range from 0 to 806,796 cases.” From the length of time communicated in the x axis description, George surmises that there is a lot of data communicated in this chart. He feels confident in his strategy to read textual descriptions instead of exploring the chart’s data-rich structures.

George presses the Tab key to focus on the interactive chart element (his screen reader switches to application mode), and he hears the screen reader announce the interactive chart as such and is prompted to press “the Enter key to start.” The prompt is followed by an announcement of the summary insight: “The chart contains 3 distinct peaks, otherwise average cases remain consistently low. The largest and steepest of the peaks occurs in January 2022, sharply reaching a maximum of 806,795 averages cases. The other two peaks are almost a quarter the size and more gradual slope. They occur in January and September of 2021.”

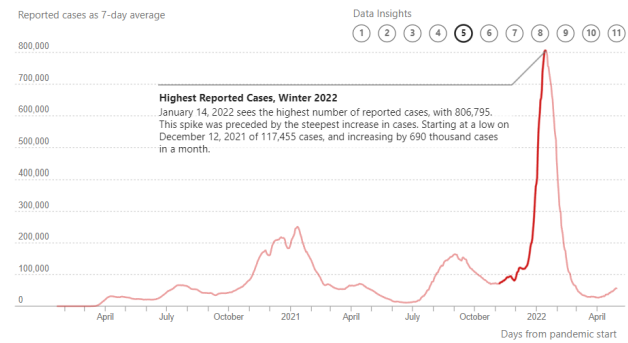


Figure 4: A line chart communicating daily reported COVID-19 cases in the United States as a 7-day rolling average. The data spans from January 21, 2020 to April 30, 2022. Currently, the trend insight communicating a surge in cases from November 2021 to January 2022 is selected.

George presses Enter to navigate into the chart’s first level. At this level, the structure consists of several regions that afford different access modalities to the chart (data insights, x axis, y axis, data points). George lands on the first region, announced as: “Data insights. There are 11 total data insights: 1 summary, 5 trends, 4 landmarks, 1 statistics.”

Data insights describe the salient aspects of a chart that might be perceived visually. They are grouped by categories, each insight type describes similar, structured information that George can anticipate. He presses Enter landing on the first category: “Summary data insights. 1 total.” George has already read the summary insight, so he presses the Right Arrow key to move over to the next category: “Trends insights. 5 total.” Going down a level, he reads the first of the trend insights: “Second Largest Peak, Winter 2021...” But George cuts off the announcement as he is interested in recent events. He presses the Right Arrow key until he reaches: “Highest Reported Cases, Winter 2022. January 14, 2022 sees the highest number of reported cases, with 806,795. This spike was preceded by the steepest increase in cases. Starting at a low on December 12, 2021 of 117,455 cases, and increasing by 690 thousand cases in a month. 4 of 5 trends.” He then presses Enter to focus on the data described by this trend insight. The screen reader announces the first data point of this trend: “117,455 average cases. December 12, 2021.” Pressing Right Arrow key, George moves to the next data point: “118,611...” He presses Right Arrow key again, cutting off the announcement and moving to the next day: “118,144...” George keeps reading values in this manner.

He then realizes an expedited way to perceive this peak would be to sonify the data. George presses Shift + Enter keys while on “December 10” and the chart plays a sonification of pitch-matched tones that represent the average cases for each day. George can hear the pitch quickly ascending, until they reach a high whistle and stops at the last data point in this trend insight. The chart announces the last day’s values: “805,062 average cases. January 15, 2022.” From the sonification, George is able to perceive the “steepest increase in cases” mentioned in the trend insight.

4.4 Experience 2: Multi-Series Line Chart

Emma comprehends best when she can slice-and-dice the data. She interrogates the data from different perspectives contained within the chart's regions that focus on sets of data at a time (e.g., aggregations, consecutive values). Emma is good at piecing together atomic information into a higher-order of understanding through her strategies. In this experience, Emma explores a multi-series line chart displaying global currency value equivalent to United States Dollars (USD) over time from 1999 to 2017 in months (Figure 5). She starts by reading the chart's description: "This a line chart with 5 data series. The series are: Brazilian Rial (BRL), New Zealand Dollar (NZD), Australian Dollar (AUD), Euro (EUR), and Great British Pound (GBP). This chart has 1 x-axis displaying time (in months). The dates range from January 1999 to December 2017. This chart has 1 y-axis displaying value (in USD). Data ranges from 0 to 2.07."

Emma then navigates to the interactive chart and reads the summary insight: "The five series in the chart follow similar trends, with the slight exception of the Brazilian Rial from 1999 to 2002. For almost all of the chart, the Pound has the highest value, followed by the Euro, [...] The most notable feature in the chart is a valley in all 5 currencies where they drop sharply in value in 2008 and then increase again in value in 2009. Aside from this valley, most changes within the chart are gradual and tend to follow years-long trends."

Emma then becomes interested in finding out more about "the valley in 2008." First, she wants to know the scale of the drop during 2008 for the currencies. Emma presses Escape twice to reach the "Data Insights" region, and then presses the Right Arrow key once to reach the X Axis. Then the axis describes itself: "X axis from 1999 to 2017. This region displays binned data points by Year as average value. There are 19 total Year bins, each with 5 data series." Emma drills down one level, arriving at the first "x axis bin" for "1999." She then moves to the right, hearing "2000, 2001, 2002, ..." as she moves along the bins. Emma stops pressing the Right Arrow key at "2008" and lets the announcement finish: "2008 average values are Brazilian Rial at 0.56, New Zealand Dollar at 0.71, Australian Dollar at 0.85, Euro at 1.47, Great British Pound at 1.85. 10 of 19 bins." She then checks the next year: "2009 average values are Brazilian Rial at 0.51, New Zealand Dollar at 0.64, Australian Dollar at 0.79, Euro at 1.39, Great British Pound at 1.57. 11 of 19 bins." Emma notices the averages go down from 2008 to 2009, but not as much as she expected.

Emma wonders if the yearly averages are smoothing out "the valley in 2008" that was described in the summary insight. She drills down within the 2009 bin. The chart announces the first data point, date in 2009, and series as "0.43 value, January 2009, Brazilian Rials series." She presses the Up Arrow key to read out the higher series value: "New Zealand Dollar series. 0.55 value, January 2009." This time the series is announced first to call attention to the change in structure. Emma navigates to the last series, "Great British Pound" and then starts traversing to the right, waiting to hear a value greater than the previous. That increase happens in "April 2009." Emma hypothesizes that "the valley" is before this date. Instead of comparing numeric values as before, she opts for comparing with sonification. Pressing Shift + Left Arrow keys, she navigates backwards by each month, hearing a tonal pitch for the "Great

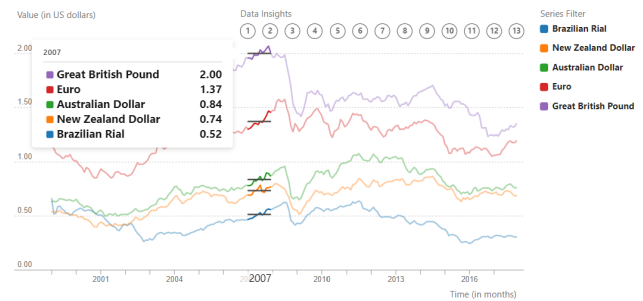


Figure 5: A multi-series line chart displaying 5 global currencies compared to the United States Dollar (USD). The chart shows the currency value equivalent in USD over time from 1999 to 2017 (in months). Currently, the x-axis bin for 2007 has been focused, displaying the averaged value in USD for each currency.

British Pound" value. Emma hears high pitches that do not change for the first 4 points, but the next 6 quickly ascend to much higher values (since navigating in reverse, the value is descending forward in time). She realizes this must be the cliff that drops into "the valley of 2008." As she expected, the "x bins" helped her navigate to this point, but the averaged values smoothed out this information.

Emma also ponders how closely currencies trends are correlated, especially since Great Britain used to be part of the European Union. She wants to filter out to compare two currencies, the "Great British Pound" and the "Euro." Emma jumps to the series filters by pressing the F key. Emma drills down and checks off "Brazilian Rial, New Zealand Dollar, Australian Dollar" and verifies that "Euro" and "Great British Pound" are checked. She navigates back to the compare series region. Here she can play spatialized sonification of both series. Emma starts by sonifying one at a time with the Shift + Right Arrow key, but it takes a while. Instead she presses Home to start over again, this time sonifying every fifth data point with the Shift + Page Down key. Emma hears time-staggered tones representing the "Euro" value in her left ear and the "Great British Pound" in her right. When she wants the values to be announced for a specific month, Emma takes a break from pressing Shift + Page Down key. After speed sonifying through the data, Emma is able to hear that the two currencies follow each other closely, as their change in pitch always seem to go up or down together. Emma also realizes that toward the end of the time period, the "Euro" closes the gap on the "Great British Pound" as their pitches get closer after "2008."

4.5 Experience 3: Stacked Bar Chart

Murphy prefers to understand data through non-speech audio. She prefers understanding data through changes in pitch and has previously created her own data sonifications with midis and synthesizers. Murphy finds that trends and distributions can be cognitively perceived through audio. In this experience, Murphy explores a stacked bar chart conveying Olympic medal counts for the top 5 countries in the Summer 2020 Olympic Games (Figure 6). She begins with the chart description: "This is a bar chart with 3 data

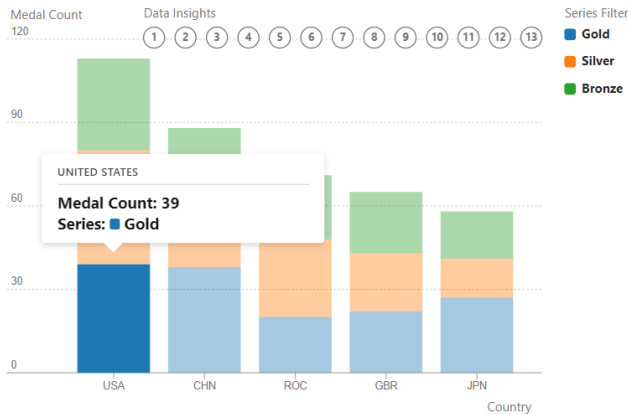


Figure 6: A stacked bar chart displaying Olympic medals won by the top 5 countries during the 2020 Summer Olympic Games. The chart shows medal type (i.e., Gold, Silver, Bronze) as colors in the stack. Currently, the data point displaying 39 Gold medals by the United States is selected.

series for Medal Type. The series are: Gold, Silver, Bronze. This chart has 1 x-axis displaying Country. The countries include from United States, China, Russia, Great Britain, and Japan. This chart has 1 y-axis displaying Medal Count. Medal counts range from 0 to 113.”

After focusing on the chart, Murphy learns from the summary insight that “the United States has the highest total number of medals, followed by China, Russia, and Great Britain. Japan has the lowest total number of medals. Of the Gold medals, the US has the highest number of medals and Russia has the lowest...” She cuts off the summary insight as enumerations are not as useful for her to make comparisons. Instead, she navigates to the “x axis” by drilling down and over one region. She wants to understand which country had the highest total medal count, and which had the least. Drilling down into the first category on the “x axis” she reads the first country’s total: “United States. Total medals are 113. 1 of 5 Countries.” and continues to read each total until “Japan”: “Japan. Total medals are 58. 5 of 5 Countries.” When she presses the Right Arrow key again, the chart emits a “Bonk” sound, indicating a boundary, from which Murphy knows she is at the end of the “x axis.”

Murphy now wants to get a sense of the scale for these totals, so she jumps back to “United States,” and then presses Shift + + Left Arrow key to play a sonification for each country. She hears three tones for each country, time-staggered and spatialized from left to right for each medal type. Murphy was expecting one tone displaying total medal count for each country, but the three tones help her perceive the distribution of each country’s medal count. She goes back to the first country’s sonification with Shift + Home key. For the “United States,” three similar high pitch tones tell her that they have won almost equal numbers of medal types. Murphy continues to interrogate the sonification for each country. For “China,” the chart emits two similar high pitch tones, and one lower pitch tone. She perceives these tones to mean “China” has as higher and almost equivalent count of “Gold” and “Silver,” but fewer “Bronze” medals.

Also, she notices that “Great Britain” has relatively equivalent tones, although all pitch levels are lower than the “United States.”

Hearing the high pitch of the “United States Gold” medal count makes Murphy wonder how their “Gold” count alone compares to the other countries combined medal counts. She jumps to the “y axis” and reads: “Y axis displaying Medal Count, from 0 to 120. This region displays stacked bars binned by increments of 30 medals. There are 4 total bins.” She drills down to the first bin: “0 to 30 medals bin contains 3 stacked bars. Gold for Russia, Great Britain, and Japan. 1 of 4 bins.” This bin does not contain “Gold for United States” so she continues: “30 to 60 medals bin contains 6 stacked bars. Gold for United States, and China. Gold plus Silver for Russia, Great Britain, and Japan. Total for Japan. 2 of 4 bins.” This is the bin she wants and it appears to have a significant number of stacked bars. Murphy then drills down to read the stacked bar descriptions and begins traversing across: “39 medals. Gold for United States. 1 of 6 stacks.” And next: “38 medals. Gold for China. 2 of 6 stacks.” Near the end: “41 medals. Gold plus Silver for Japan. 5 of 6 stacks.” And “43 medals. Gold plus Silver for Great Britain. 4 of 6 stacks.” Murphy learns that the “United States” almost won as many “Gold” medals as “Gold” and “Silver” for both “Great Britain” and “Japan.”

5 FINDINGS: EVOLUTION OF ACCESSIBLE VISUALIZATION EXPERIENCES

In this section, we present the findings of our parallel co-design study as vignettes of how our accessible visualization designs evolved throughout the process. Each vignette describes a guideline or lesson that we learned and can be applied when creating accessible visualization experiences for other chart types. We provide insights into the ideas proposed by our design partners, their feedback after contextually interacting with the concepts, and subsequent confirmation, refinement, or dismissal of the design. To catalogue the vignettes, we have indexed each by their combined design dimensions: *structure, navigation, description, non-speech audio, and focus* (see Table 1 for descriptions).

Orthogonal, varied access to information. Structure & Navigation

The design goal of our structure and navigation was to provide a variety of access points to the chart’s information and underlying data. This decision was based on accessible design practice around understanding options and customizability: users will have varying mental models of the application and expect to be able to approach information from different directions. For session 1, we adopted a hierarchical tree structure similar to that proposed by Zong *et al.* [38]. Structure makes the complexity of a chart manageable, as P7 notes in session 1: “Charts, just like any other controls ... can get a little complex based on the number of elements that you’re trying to interact with. And I feel like the pattern of navigation ... to expand and collapse and then the ability to get back to a particular level pretty quickly. I think that is also something that really worked in this case.”

Our design partners each developed their own styles of exploring the data, based on what best suited their informational style or what aligned with their mental models. P6 commented in session 2 that “one of [the design’s] strengths is the variety of ways to look at the data. That’s probably the the biggest strength. It’s a little bit like having a zoom-in and zoom-out model.” Similarly, P1 remarked that “the

whole structure in the prototype is very useful to me, because it lets me navigate to the desired granularity.”

As we iterated on accessible visualization designs, we did not detour from this initial hierarchical tree structure. In fact, we continued to build out the granularity of the chart’s structure. For example, in session 2, we included an additional layer within the y-axis bins, segmented by each consecutive timespan that fell within the bin. Later in session 3, we extended the y-axis bins for stacked bar charts: binning partially stacked bars (i.e., partial sums). P6 found the stack comparisons useful: *“The concept of [not only] comparing totals, but also comparing proportions as well; that made a lot of sense to me.”* As our design partners got more comfortable with traversing the tree structure, they asked for advanced navigation patterns such as lateral movements across branches (e.g., from the x-axis to y-axis).

Interoperability between descriptions and sonification. *Navigation, Description, & Non-Speech Audio*

We put an emphasis on enabling SRUs to consume data by their preferred medium. When navigating with a screen reader, focusing on an element initiates an announcement of that element’s description label (i.e., *aria-label*). Beyond speech-based descriptions, sonification has been found to be useful for people to perceive temporal changes in values, value comparison, and value distribution. Therefore, we implemented sonification for the initial prototype as uninterrupted playback of the current focused element’s data (e.g., targeted data of an insight). Our design partners had an expectation that the navigation would control sonification playback: *“I was a little confused when I was [sonifying] because my assumption was that it would continue to beep as I arrowed across.”* (P2, session 1). P7 envisioned the following solution: *“imagine if I can read the data points quickly, but sonification is gonna be even faster. I can just play it as a way to like, navigate, find interesting places and stop here. And then I can look around.”* Our design partners also explained that the sonification lacked context for what values were being sonified. They conceptualized that the data value should be announced whenever the sonification pauses.

In session 2, we modified navigation with the *Shift* key, to emit a sonification on the navigated element, instead of a textual description. P2 described the experience as *“I’m moving through this data and right now as I move left to right, hearing the sonifications, which I quite like. It just feels a lot faster to me than [reading] the values.”* Whenever the sonification stops, the screen reader announces the description of the previously sonified data point. P5 confirmed *“comprehension really improves because of those individual sonifications.”* Later we tried this approach on other chart elements that contain sets of data values (e.g., x bins). Our design partners found the spatialized, burst of tones representing a multiple data points to be helpful: *“I wasn’t sure how I would use sonification, but I feel like it makes sense to have that option to quickly perceive something that would take a long time to read the data.”* (P4, session 3). Sonification mixed with descriptions, however, were not always the best option. In session 2, we designed the sonification for multiple-series lines to play out as interspersing sonification with descriptions of the start and end data points. Design partners found it difficult to follow, as they would rather sonify and navigate individual data points.

Orientation and signposts. *Structure, Navigation, & Non-Speech Audio*

As our design partners navigated through our prototype, we quickly found that the complexity of our structure required that we provide better guidance to help our users orient themselves and know what locations were available to them from a given point. Our first attempt at this involved reducing the verbosity of data descriptions: by removing redundant information (for instance, what series was being explored when navigating within a series), we hoped users would more quickly glean information about where they were. This change had a mixed reception: while generally useful in supporting faster navigation, there were times when our partners did still want the full data, and it still did not do enough to quickly tell them where they could or could not go.

To address the issue of knowing where users could or could not go, we introduced a “boundary tone” (in session 3), a noise earcon distinct from existing sonification sounds that indicated when users could not navigate in the direction they were attempting to. With the inclusion of this tone, users could also navigate more confidently in the absence of this tone, reassured that they were moving in a valid direction. This subsequently influenced the structure of the prototype, as we learned that having boundaries where this tone played was actually more useful for navigation and orientation than having “wrap around,” which we had initially implemented based on perceived convenience.

The last feature to evolve here was our batch of hotkeys. Inclusion of the *Home* and *End* keys as shortcuts to the first or last element on a level allowed our partners to locally orient themselves, as well as navigate more quickly. To provide one more signpost for orientation, as well as an escape in case users got lost or stuck, we then implemented semantically meaningful keyboard shortcuts to jump directly to major points within the structure (e.g., *X* to jump to the *X*-Axis, *F* for the Filters).

Enabling comparisons. *Structure, Non-Speech Audio, & Focus*

A challenge that emerged as we moved from single-series data to multi-series data was the task of comparison. Our initial multi-series support allowed users to interact with all the data from each series, and we hoped that the ability to compare adjacent values across series would enable comparison across them. But one design partner (P1) in session 2 commented *“if I wanted to understand the relationship between two different series, that would become a little more complex. Right now this feels like two line charts on top of each other.”* This prompted us to explore more explicit modes of simultaneous comparison. P9 recommended *“stereo, where one series plays through one ear and [the other series] plays through the other ear.”* This feature allows for simultaneous comparison of series, while using spatialization to distinguish between values and allow for persistent identification of a given series. The next feature implemented was filtering: by allowing users to filter out the series they were not interested in comparing, we minimized the auditory and processing loads on the user and enabled additional customization of the user experience. Both features proved to complement each other well, as spatial audio beyond two line series was difficult to perceive: *“I think I could make out two [spatialized tones]. I couldn’t make out the middle one.”* (P1, session 3).

External consistency. *Structure, Navigation, & Focus*

Throughout the design process, we worked with our design partners to explore how to make our visualization controls consistent with external conventions learned in other applications. Per P4's advice, "Keep in mind ... much depends on somebody's ability, not only to use their tools. But their familiarity with stuff makes a difference..."

Some of our initial controls matched conventions nicely, such as the use of the *Left* and *Right Arrows* for basic movement within a level, and the use of *Shift* strictly as a modifier key paired with other inputs. Other controls did not match expectations in some contexts, such as the use of the *Left* and *Right Arrows* to navigate between checkboxes in our *Filters* section: in other applications, *Tab* is most frequently used for this purpose, and our partner with low vision expected the *Up* and *Down Arrows* to navigate between them, as the filters were visually laid out in a vertical list. In this case, we plan to implement redundant controls when possible, to match as many expectations as possible.

However, sometimes design partners' expectations diverged from each other in places that can only have one control, such as in the use of *Enter* and *Escape* to navigate between levels: most of our partners said this matched their expectations, but one partner reported frequent confusion as they expected the *Up* and *Down Arrows* to perform that function. In this case, we plan to maintain our initial *Enter/Escape* model, with the recognition that an option to customize keyboard inputs would allow for better alignment with learned experience.

The remaining domain that relied heavily on user expectation was sonification: our visualization prototype uses a fixed pitch scale, with a base tone at the bottom of the Y-axis and a logarithmic pitch scale hitting a maximum value at the top of the Y-axis. With this design, pitch correlates to the percent height up the Y-axis, rather than an absolute value. Although we were concerned users might not understand how scales varied between different charts, we found this was not a problem. When one partner was asked to guess how far up the Y-axis they were based on a given tone, they accurately guessed purely based on expectation. However, due to some uncertainty in the guess, we plan to include a reference for the scale in future versions of the tool.

Describe insights, not annotations. Structure & Description

An important aspect of our design was to describe salient aspects of the chart in approachable prose for BLVIs. We learned from our design partners that specific details matter: standard "framing" or "templating" helps to anticipate information, and verbose descriptions are good, when warranted. In session 1, our first attempt to describe insights resembled annotations from visual charts. They were short and varied in structure as P5 noted: "*The [insights] seem to follow a format, but don't exactly. It would be great if they did [...] follow the format [consistently].*" P5's solution was "*a predictable format which I know, this is how an [insight] usually goes.*" To assist in navigating the insights, we also grouped them by "*insight type*," supporting users to access the specific category of information they are searching for.

We also learned to base descriptions on tangible objects, rather than intrinsically visual things (e.g., steep inclines, valleys). Finally, we learned well-crafted descriptions can paint a picture for BLVIs: "*I love beautiful pictures [...] I have a guide dog and they think he's so beautiful and they'll describe him to me. That's wonderful. When*

I see him, I see one part of his whole body, his head or his harness or whatever. I don't see the whole picture. So, when somebody paints that picture verbally, that's amazing to me." (P3, session 1).

Reading vs. Announcements. Description & Focus

Interacting with a chart requires a screen reader user to change the mode in which they typically navigate web page elements (e.g., *turning virtual cursor off* in JAWS or *application mode* in NVDA). This context switch is necessary for the chart to be interpreted by the screen reader as a web application, passing keystrokes that would otherwise serve specific functions for screen reader operation. While this mode is necessary to navigate complex structures, it does not allow users to read text as they normally would. Each chart element's description is announced once, unless interrupted. In session 1, partners found it difficult to remember parts of a long description. They also found it time consuming to re-listen to the entire description again. P5 commented "*I want to be able to review those sentences individually or go read them by words [or] characters. Maybe there's something I missed. I want to read it carefully.*"

To address this limitation, P5 recommended we introduce "*document mode*." This new mode enables users to read a chart element's description in a focused mode with all of the reading interactions typically available to standard web elements. Our design partners found this feature helpful to interpret large blocks of text, although some of them found it clumsy to operate, as they would have to manually switch screen reader modes (i.e., *turn virtual cursor on*). This feature also spurred our partners to conceive of how the chart would be read in browse mode: they envisioned the insight categories as headers, and individual insights as paragraphs within. Each insight could then link directly to landmarks in the interactive version of the chart, supporting users to transfer between browsing and interactively exploring.

Random access to data. Structure & Focus

In accessible design, when dealing with complexity or volume it is helpful to lighten the cognitive load by focusing on only the relevant content. Backed by data that users can query, visualizations inherently provide opportunities to refocus based on a subset of data. For example, in session 3 we added filtering at the behest of our design partners: "*The filtering ... that is phenomenal. That's gonna make everybody real happy. Being able to go in and take out the things that are not being used ... If I don't have as much to listen to, then I'm going to come to a conclusion, good or bad, right or wrong, much quicker.*" (P3, session 3).

From an implementation perspective, when a user filters out a series, Chart Reader updates the chart's visual and descriptive elements. This works well for any chart-generated content such as data glyphs (e.g., line series) or chart elements (e.g., y axis bins) but is ambiguous for author-generated content (e.g., insights). In our implementation we removed annotations from the structure if all of their described series were filtered out. Complications arise if filtering along the x or y axis is supported, as insights might target partially available data.

Our design partners also desired the capability to focus on subsets of data by user-defined queries. For example, custom binning along either x or y axis would be useful to describe a summary of the selection and access to the queried data. Design partners mentioned the default bins were "*arbitrary*" based on the author's

decision. They would rather specify their own bins, especially in cases where there were contextually important values to bin by. Partners also ideated on querying data by statistical features (e.g., “*find where these two lines intersect*” (P2), “*find the nearest peak*” (P7)). Our design partners also envisioned random access to the structure of the chart. Instead of orienting based on the descriptions of chart elements, the chart would describe the user’s current location within the structure and with respect to nearby points of interest.

6 DISCUSSION AND FUTURE OPPORTUNITIES

6.1 Value of Autonomy in Data Visualization Consumption

A major goal of our design was to provide autonomy and control to BLVIs to engage as much as they want with the content of a data visualization. It is motivated by the notable inequity in autonomy that BLVIs have when engaging with a visualization compared to other users, who are afforded the opportunity to quickly glance at a chart and focus in on what is interesting to them (if anything) before concluding their engagement with the chart. On the other hand, BLVIs are often given only shallow alt-text of a visualization that does not enable deeper engagement, or raw data that demands longer or more complex interaction than potentially desired.

With this goal in mind, we were excited to see our design partners develop their own personal styles of engaging with charts throughout our study, reflecting their autonomy in deciding how to consume visualizations with our tool. Most notably, our hierarchical “dig deeper” model of data structuring allowed each partner to control how deeply they engaged with each chart’s content. Starting at the high-level summary insight, some partners dug all the way down to the raw data to conduct their own exploration and analysis, while others listened to the summary insight and explored a few other insights before concluding engagement. The many directions from which a user can approach the data (e.g., from a multi-series comparison vs. from the summarized insights) further supports this autonomy.

Supporting this style of autonomous chart exploration is particularly important as we consider real-world environments where users might encounter charts, such as news articles or research papers. In these cases, users may have vastly differing goals in what they want to achieve in their interaction with a chart; we believe our tool will support this and hope to confirm this in the future with more ecologically valid studies.

6.2 Opportunities & Challenges in Designing Web-Based Screen Reader Accessible Tools

One consistent design priority of our accessible visualization was to integrate into users’ existing work flows as much as possible, which meant ensuring users could use their own screen readers with personalized configurations, and that they do not need to acquire and learn additional hardware. This led us first to our web-based prototype, which could be integrated into existing workflows of news websites, social media, or other Web platforms that might integrate charts into their content. A natural affordance of this

integration was that users could begin interaction with our visualization immediately. Additionally, we strove to mimic standard screen reader web interaction with our keyboard controls (such as, including the hotkeys Home and End to jump to the start or end of a list, reserving certain keys like Shift as modifier keys).

However, challenges emerged in how various screen readers controlled or modified input when interacting with a Web-based application. For example, as we strove to minimize unnecessary speech, we fought with NVDA saying “selected” anytime a shift key was held down; and certain keyboard inputs, such as pressing Caps Lock, were silently handled by the screen reader.

From the web development perspective, it was also difficult to identify certain screen reader metadata that was important to aspects of desired accessible visualization interactions. When attempting to simulate timing, for instance, no metadata was available regarding the speed at which the screen reader was speaking, making it impossible to properly time certain actions that relied on waiting for the screen reader to fully announce an element.

With this in mind, we point to development opportunities within the screen reader and Web design spaces, as finding methods to integrate functionality of the two domains could enable better accessibility in future web-based applications.

6.3 Reflections on Our Parallel Co-Design Method

Though unconventional, our method of parallel and iterative co-design was highly effective and informative. One of the biggest advantages of this model was the flexibility of our co-design sessions. Given that a frequent deterrent to people with disabilities participating in research is accommodation challenges, accommodating participants’ schedule changes was an asset to encourage their participation, it also made them feel respected and valued as design partners.

Compared to group co-design sessions where many design partners collaborate on a single design, our model allowed us to develop multiple designs from a single prompt (i.e., iteration of our prototype) and identify what elements of the design were more common across various designs, and what conflicting design elements might be better attributed to personal preference. This also enabled us to focus more on the parts of the design that each participant was passionate about or had more experience with, as we could be confident that all the various parts would be addressed in some detail across the 9–10 design sessions with each prototype.

The iterative model (which is more common in co-design practice) was also effective in this process, as it removed pressure to design or implement features on-the-fly with partners, and allowed our sessions to be more driven by understanding partners’ experiences with the prototypes than by abstract design. In addition, as the study progressed, each iteration of the artefact improved by branching out to a new modality and refining previous concepts, while keeping what worked well. Besides the desired design ideas, we learned what did and did not work in the iterative sessions. Furthermore, our design partners were pleased to see their feedback incorporated into the revision. For example, in session 2, P2

commented: *“I’m honestly very impressed about the amount of feedback you got and the amount of innovation you took based on that feedback, that’s excellent. Like that’s just tremendously great to see.”*

However, our co-design process had its hiccups. For instance, when a design partner’s preference would not be aligned with how a specific feature should be implemented or a problem be addressed, we were sometimes forced to overrule their preference, without the ability to discuss it with them before its implementation in a future session. Though we strove to ensure all participants had at least one of their requested features integrated into each new prototype, some participants did note that we chose not to pursue their design concept without a clear understanding of why we made that choice.

Additionally, given that each session’s design prompt built on the outcome of the prior session’s design changes, missing a design session led to a disconnect at the following session. Since the new design already was implemented, it did not make sense to elicit feedback on the prior version of the prototype, but then it became harder for the partner to feel as connected to or invested in the newer prototype.

6.4 Limitations

A limitation of our accessible visualization rendering engine is our predominant focus on blind users. Our approach gave less consideration to other people with disabilities who might use a screen reader, such as low vision individuals or people with dyslexia or visual processing disorders. As we found through working with our one low-vision partner, being able to complement screen reader description with corresponding visuals can be an important element of an accessible interaction for low vision individuals. However, our design almost exclusively focuses on audio representation with little supplementary visuals. Additionally, our prototype is more effective when used by people with a strong familiarity with screen readers: individuals who are less experienced with screen readers may have a harder time operating them.

Another limitation of our study is the controlled context we provided for our chart interactions, most notable in how we determined the datasets for each session. We strove to provide charts on topics our partners would find interesting. However, there were a few occasions where design partners mentioned that they were not particularly interested or invested in the dataset, even though we did not have any instances where our partners did not understand the data. Some partners commented that they felt they would interact with it more actively if it were data they cared more about. With this limitation in mind, we recommend that future studies provide the opportunity for participants to provide the data they care about or to share topics they are interested in to inform the selection of study material.

Finally, our study did not go as planned in certain respects due to the time constraints created by the need to implement new features in our prototype between co-design sessions. To implement new features, we needed to transcribe the recorded session videos, code the transcripts, generate designs from the codes, and then implement the designs—a largely sequential process that posed a significant obstacle to our small research team. As a result, we ended up reducing the number of sessions we could run, as our study was constrained to a specific time window (five months) and

the increased workload of implementing features between sessions spaced out the sessions more than initially expected. Due to this increased spacing between sessions, there were some cases where partners had forgotten certain elements of the earlier prototypes. Although we attempted to preempt this by providing partners the ability to freely interact with the tool between sessions, few participants availed this opportunity, which required that we spent more time in sessions reviewing controls; this likely altered partners’ interaction with the tool during the session.

6.5 Future Research Opportunities in Realizing Better and Broader Data Experiences

One logical next step is to evaluate our accessible visualization designs with a larger group of BLVIs. In this section, we discuss several research opportunities to realize better and broader data experiences, derived from the nuanced and thoughtful feedback from our partners who have varying levels of expertise and experiences.

Designing Tutorials for Accessible Visualizations. Our design partners loved the capability and flexibility to interactively read and understand visualizations and their underlying data. The high level of flexibility, however, comes at the cost of simplicity. Our accessible visualization designs require the understanding of key features and underlying navigation model, as well as the use of many key combinations (i.e., hotkeys). During our co-design sessions, we walked our partners through new features or visualizations when they were introduced. This is not feasible in the real world setting, and providing a link to the table of keyboard interactions would not be enough to help a new BLV user become familiar with our accessible visualization designs. One of the important open research questions is how to teach BLVIs to learn and understand the underlying navigation and interaction model to effectively explore accessible visualizations. In addition, as we expand the coverage of chart types, we need a mechanism to introduce new chart types to BLVIs. Many of our design partners were not familiar with stacked bar charts, and thus we had to explain how they were drawn.

Next Steps for a Visualization Accessibility Engine. Chart Reader is a first step toward synthesizing accessible visualization features into an accessibility engine. Authors can create accessible web visualizations with Chart Reader by inputting their own data, insights, and configuration described in Section 4.1. The Chart Reader engine currently supports rendering three chart types: single-series line chart, multi-series line chart, stacked bar chart. Future design and implementation work is needed to support additional charts (e.g., scatter-plots, area charts, histograms) and visualizations with relational or geographic data (e.g., node-link diagrams, choropleth maps). Our research goal was not to develop a fully featured authoring toolkit, but to synthesize the state-of-the-art work [29, 30, 38] into a cohesive accessible visualization experience. The lessons learned and resulting features of Chart Reader could be of use in other visualization types. For example, binning of the x- and y- axes in scatter-plots was previously found to be helpful [38], or hierarchical navigation for relational data in node-link diagrams. However, it is unclear if all features would translate to other visualization types. For example, the Chart Reader sonification approach relies on sequential ordering of data – how would

readers follow along with a sonification in geographic maps without a well-defined sequence of the data points? One would surmise that the mixed speech and sonification approach of Chart Reader would help readers orient themselves when playing sonifications, but more research is needed to validate against other chart types.

Authoring Accessible Visualization Experiences. We believe that, among the six major approaches to construct data visualizations [8], textual programming (e.g., d3, Vega-Lite), template editors (e.g., Microsoft Excel), and shelf configuration (e.g., Microsoft Power BI, Tableau) are common ways to create visualizations. For textual programming, toolkits and frameworks with accessible visualization capabilities would make the construction process easier for authors. However, much of the onus is still on authors to create accessible features into charts. Integrating accessible visualization features into GUI-based authoring tools (e.g., Excel, Power BI, Tableau) would lower the burden from authors and could promote the creation of accessible visualizations. The design and development of our accessibility engine leads us towards a view of generalizing how chart components and elements can be labeled for better screen reader navigation. By understanding how components should describe themselves and the methods for navigating between and within them, we can move toward an actual practice of generalized accessible visualization experiences.

In particular, data insights, one of the accessible visualization features we designed with design partners, demand further research for practical use and adoption. While some types of data insight can be created automatically from templated sentences, others need a natural language generation algorithm tailored to different types of insights, or well-defined guidelines for human authors.

Making Data and Visualization Accessible. Data visualization is commonly used to convey meaningful insights gained from data, or to represent data in a visual form to leverage human visual capabilities. Due to its power and prevalence as a communication medium, visualization research and practice has recently started to put much effort in making data visualization on the Web accessible with screen readers, a commonly used assistive technology for BLVIs. This will be essential in helping all people effectively collaborate and communicate with data visualization, which is an open and promising research opportunity.

While the genesis of visualization stems from visual representations and abstraction of data, we note that data visualizations themselves transcend the visual medium to provide readers with an non-linear navigation experience. By their nature, screen readers serialize geometrically-positioned elements in a 2-dimensional space. While the visual position of chart components are important for layout, several of our partners noted that the notion of x- and y-axis were new or meaningless. The findings from our work inform future directions of designing these data experiences, and hone in on the importance of multiple media of communication (such as non-voice audio and sonifications), elevation of written insights with references to data elements, and supporting multiple avenues of navigation. For example, instead of providing the entire data table to the users, future tools could use the most appropriate data representation depending on the insight types, such as sonification for numerical trends, a derived table generated from appropriate data transforms, etc. This would play an important role in enabling

BLVIs to effectively analyze and explore data to understand and identify insights with individual agency.

7 CONCLUSION

In this work, we synthesized a web-based accessibility engine, called Chart Reader, which enables the generation of accessible visualization experiences compatible with screen readers. We presented accessible visualization experiences for three chart types—single-series line chart, multi-series line chart, and stacked bar chart—designed with 10 screen reader users over five months through an iterative co-design study. Combining the designs of the state-of-the-art work [29, 30, 38], Chart Reader incorporates narration with non-speech audio (e.g., sonification, earcon) and provides multiple ways to navigate chart components. We described how our accessible designs evolved throughout our process, along with a summary of lessons we learned that can be applied when creating accessible visualization experiences for other chart types. We also reflected on our co-design method and the limitations with our work. Finally, we discussed challenges and opportunities in designing web-based screen reader accessible tools and realizing new opportunities for data and visualization experiences.

REFERENCES

- [1] Apple. 2022. Apple Audio Graphs. https://developer.apple.com/documentation/accessibility/audio_graphs. (Accessed on Sept. 4, 2022).
- [2] Alice Boxhall, James Craig, Dominic Mazzoni, and Alexander Surkov. 2022. The Accessibility Object Model (AOM). <https://wicg.github.io/aom/>. (Accessed on Sept. 4, 2022).
- [3] Stephen Brewster. 2002. Visualization tools for blind people using multiple modalities. *Disability and Rehabilitation* 24, 11-12 (2002), 613–621. <https://doi.org/10.1080/09638280110111388>
- [4] Pramod Chundury, Biswaksen Patnaik, Yasmin Reyazuddin, Christine Tang, Jonathan Lazar, and Niklas Elmquist. 2021. Towards Understanding Sensory Substitution for Accessible Visualization: An Interview Study. *IEEE Transactions on Visualization and Computer Graphics* 28, 1 (2021), 1084–1094. <https://doi.org/10.1109/TVCG.2021.3114829>
- [5] Frank Elavsky, Cynthia Bennett, and Dominik Moritz. 2022. How accessible is my visualization? Evaluating visualization accessibility with Chartability. *Computer Graphics Forum* 41, 3 (Jun 2022), 57–70. <https://doi.org/10.1111/cgf.14522>
- [6] Danyang Fan, Alexa F. Siu, Hrishikesh V. Rao, Gene S-H Kim, Xavier Vazquez, Lucy Greco, Sile O'Modhrain, and Sean Follmer. 2022. The Accessibility of Data Visualizations on the Web for Screen Reader Users: Practices and Experiences During COVID-19. *ACM Transactions on Accessible Computing* (Aug 2022). <https://doi.org/10.1145/3557899> Just Accepted.
- [7] Sarah L. Fossheim. 2022. How (not) to make accessible data visualizations, illustrated by the US presidential election. <https://fossheim.io/writing/posts/accessible-dataviz-us-elections/>. (Accessed on Sept. 4, 2022).
- [8] Lars Grammel, Chris Bennett, Melanie Tory, and Margaret-Anne Storey. 2013. A Survey of Visualization Construction User Interfaces. In *EuroVis - Short Papers*, Mario Hlawitschka and Tino Weinkauff (Eds.). The Eurographics Association. <https://doi.org/10.2312/PE.EuroVisShort.EuroVisShort2013.019-023>
- [9] Sabine Hennig, Fritz Zobl, and Wolfgang W. Wasserburger. 2017. Accessible Web Maps for Visually Impaired Users: Recommendations and Example Solutions. *Cartographic Perspectives* 88 (Nov 2017), 6–27. <https://doi.org/10.14714/CP88.1391>
- [10] Highcharts. 2022. Accessibility module feature overview. <https://www.highcharts.com/docs/accessibility/accessibility-module-feature-overview>. (Accessed on Sept. 4, 2022).
- [11] Leona M Holloway, Gagatay Goncu, Alon Ilisar, Matthew Butler, and Kim Marriott. 2022. Infsonics: Accessible Infographics for People Who Are Blind Using Sonification and Voice. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) (CHI '22). Association for Computing Machinery, New York, NY, USA, Article 480, 13 pages. <https://doi.org/10.1145/3491102.3517465>
- [12] Shakila Cherise S Joyner, Amalia Riegelhuth, Kathleen Garrity, Yea-Seul Kim, and Nam Wook Kim. 2022. Visualization Accessibility in the Wild: Challenges Faced by Visualization Designers. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) (CHI '22). Association for Computing Machinery, New York, NY, USA, Article 83, 19 pages. <https://doi.org/10.1145/3491102.3517630>

- [13] Crescentia Jung, Shubham Mehta, Atharva Kulkarni, Yuhang Zhao, and Yea-Seul Kim. 2022. Communicating Visualizations without Visuals: Investigation of Visualization Alternative Text for People with Visual Impairments. *IEEE Transactions on Visualization and Computer Graphics* 28, 1 (2022), 1095–1105. <https://doi.org/10.1109/TVCG.2021.3114846>
- [14] N. W. Kim, S. C. Joyner, A. Riegelhuth, and Y. Kim. 2021. Accessible Visualization: Design Space, Opportunities, and Challenges. *Computer Graphics Forum* 40, 3 (2021), 173–188. <https://doi.org/10.1111/cgf.14298>
- [15] Richard E. Ladner. 2015. Design for User Empowerment. *Interactions* 22, 2 (feb 2015), 24–29. <https://doi.org/10.1145/2723869>
- [16] Susie Lu. 2022. D3-Annotation. <https://d3-annotation.susielu.com/>. (Accessed on Dec. 7, 2022).
- [17] Alan Lundgard, Crystal Lee, and Arvind Satyanarayan. 2019. Sociotechnical Considerations for Accessible Visualization Design. In *2019 IEEE Visualization Conference (VIS)*. IEEE, New York, NY, USA, 16–20. <https://doi.org/10.1109/VISUAL.2019.8933762>
- [18] Alan Lundgard and Arvind Satyanarayan. 2022. Accessible Visualization via Natural Language Descriptions: A Four-Level Model of Semantic Content. *IEEE Transactions on Visualization and Computer Graphics* 28, 1 (2022), 1073–1083. <https://doi.org/10.1109/TVCG.2021.3114770>
- [19] Jennifer Mankoff, Gillian R. Hayes, and Devva Kasnitz. 2010. Disability Studies as a Source of Critical Inquiry for the Field of Assistive Technology. In *Proceedings of the 12th International ACM SIGACCESS Conference on Computers and Accessibility (Orlando, Florida, USA) (ASSETS '10)*. Association for Computing Machinery, New York, NY, USA, 3–10. <https://doi.org/10.1145/1878803.1878807>
- [20] Kim Marriott, Bongshin Lee, Matthew Butler, Ed Cutrell, Kirsten Ellis, Gagatay Goncu, Marti Hearst, Kathleen McCoy, and Danielle Albers Szafr. 2021. Inclusive Data Visualization for People with Disabilities: A Call to Action. *Interactions* 28, 3 (apr 2021), 47–51. <https://doi.org/10.1145/3457875>
- [21] Microsoft. 2022. Overview of accessibility in Power BI. <https://docs.microsoft.com/en-us/power-bi/create-reports/desktop-accessibility-overview>. (Accessed on Sept. 4, 2022).
- [22] Microsoft. 2022. Screen reader support for Excel. <https://support.microsoft.com/en-us/office/screen-reader-support-for-excel-0976b140-7033-4e2d-8887-187280701bf8>. (Accessed on Sept. 4, 2022).
- [23] Priscilla Moraes, Gabriel Sina, Kathleen McCoy, and Sandra Carberry. 2014. Evaluating the Accessibility of Line Graphs through Textual Summaries for Visually Impaired Users. In *Proceedings of the 16th International ACM SIGACCESS Conference on Computers and Accessibility (Rochester, New York, USA) (ASSETS '14)*. Association for Computing Machinery, New York, NY, USA, 83–90. <https://doi.org/10.1145/2661334.2661368>
- [24] Alan F. Newell and Peter Gregor. 2000. “User Sensitive Inclusive Design”—in Search of a New Paradigm. In *Proceedings on the 2000 Conference on Universal Usability (Arlington, Virginia, USA) (CUU '00)*. Association for Computing Machinery, New York, NY, USA, 39–44. <https://doi.org/10.1145/355460.355470>
- [25] SAS. 2022. SAS Graphics Accelerator. <https://support.sas.com/software/products/graphics-accelerator/index.html>. (Accessed on Sept. 4, 2022).
- [26] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2017. Vega-Lite: A Grammar of Interactive Graphics. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 341–350. <https://doi.org/10.1109/TVCG.2016.2599030>
- [27] Ather Sharif, Sanjana Shivani Chintalapati, Jacob O. Wobbrock, and Katharina Reinecke. 2021. Understanding Screen-Reader Users’ Experiences with Online Data Visualizations. In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility (Virtual Event, USA) (ASSETS '21)*. Association for Computing Machinery, New York, NY, USA, Article 14, 16 pages. <https://doi.org/10.1145/3441852.3471202>
- [28] Ather Sharif, Olivia H. Wang, and Alida T. Muongchan. 2022. “What Makes Sonification User-Friendly?” Exploring Usability and User-Friendliness of Sonified Responses. In *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility (Athens, Greece) (ASSETS '22)*. Association for Computing Machinery, New York, NY, USA, Article 45, 5 pages. <https://doi.org/10.1145/3517428.3550360>
- [29] Ather Sharif, Olivia H. Wang, Alida T. Muongchan, Katharina Reinecke, and Jacob O. Wobbrock. 2022. VoxLens: Making Online Data Visualizations Accessible with an Interactive JavaScript Plug-In. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) (CHI '22). Association for Computing Machinery, New York, NY, USA, Article 478, 19 pages. <https://doi.org/10.1145/3491102.3517431>
- [30] Alexa Siu, Gene S-H Kim, Sile O'Modhrain, and Sean Follmer. 2022. Supporting Accessible Data Visualization Through Audio Data Narratives. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) (CHI '22). Association for Computing Machinery, New York, NY, USA, Article 476, 19 pages. <https://doi.org/10.1145/3491102.3517678>
- [31] Desmos Studio. 2022. Graphing Calculator. <https://www.desmos.com/calculator>. (Accessed on Sept. 4, 2022).
- [32] Tableau. 2022. Accessibility: Putting the power of the Tableau Platform into the hands of everyone. <https://www.tableau.com/products/accessibility>. (Accessed on Sept. 4, 2022).
- [33] VISA. 2022. VISA Chart Components. <https://developer.visa.com/pages/chart-components>. (Accessed on Sept. 12, 2022).
- [34] W3C. 2022. WAI-ARIA 1.1 spec. <https://www.w3.org/TR/wai-aria/>. (Accessed on Sept. 4, 2022).
- [35] W3C. 2022. WAI-ARIA “Complex Images” tutorial. <https://www.w3.org/WAI/tutorials/images/complex/>. (Accessed on Sept. 4, 2022).
- [36] W3C. 2022. WAI-ARIA Graphics spec. <https://www.w3.org/TR/graphics-aria-1.0/>. (Accessed on Sept. 4, 2022).
- [37] W3C. 2022. Web Audio API. <https://www.w3.org/TR/webaudio/>. (Accessed on Sept. 4, 2022).
- [38] Jonathan Zong, Crystal Lee, Alan Lundgard, JiWoong Jang, Daniel Hajas, and Arvind Satyanarayan. 2022. Rich Screen Reader Experiences for Accessible Data Visualization. *Computer Graphics Forum* 41, 3 (2022), 15–27. <https://doi.org/10.1111/cgf.14519>