Azimuth: Designing Accessible Dashboards for Screen Reader Users

Arjun Srinivasan
Tableau Research
Seattle, WA, USA
arjunsrinivasan@tableau.com

Tim Harshbarger*
Deque Systems
Bloomington, IN, USA
tim.harshbarger@deque.com

Darrell Hilliker
Salesforce.com
Tempe, AZ, USA
dhilliker@salesforce.com

Jennifer Mankoff†
University of Washington
Seattle, WA, USA
jmankoff@cs.washington.edu

ABSTRACT

Dashboards are frequently used to monitor and share data across a breadth of domains including business, finance, sports, public policy, and healthcare, just to name a few. The combination of different components (e.g., key performance indicators, charts, filtering widgets) and the interactivity between components makes dashboards powerful interfaces for data monitoring and analysis. However, these very characteristics also often make dashboards inaccessible to blind and low vision (BLV) users. Through a co-design study with two screen reader users, we investigate challenges faced by BLV users and identify design goals to support effective screen reader-based interactions with dashboards. Operationalizing the findings from the co-design process, we present a prototype system, Azimuth, that generates dashboards optimized for screen reader-based navigation along with complementary descriptions to support dashboard comprehension and interaction. Based on a follow-up study with five BLV participants, we showcase how our generated dashboards support BLV users and enable them to perform both targeted and open-ended analysis. Reflecting on our design process and study feedback, we discuss opportunities for future work on supporting interactive data analysis, understanding dashboard accessibility at scale, and investigating alternative devices and modalities for designing accessible visualization dashboards.

CCS CONCEPTS

• **Human-centered computing → Visualization**; **Accessibility**; **Human computer interaction (HCI)**.

KEYWORDS

Dashboards, screen readers, text generation

[*] Author was affiliated with Salesforce.com as an Accessibility Consultant at the time of this work.
[†] Author was a visiting researcher at Tableau at the start of this work.

ASSETS '23, October 22–25, 2023, New York, NY, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0220-4/23/10.
https://doi.org/10.1145/3597638.3608405

ACM Reference Format:
Arjun Srinivasan, Tim Harshbarger, Darrell Hilliker, and Jennifer Mankoff. 2023. Azimuth: Designing Accessible Dashboards for Screen Reader Users. In *The 25th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '23), October 22–25, 2023, New York, NY, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3597638.3608405

1 INTRODUCTION

The inherent graphical nature of data visualizations has historically made them inaccessible to people who have blind or low vision (BLV)[1]. However, the growing popularity of visualizations on the web along with legal mandates for making web-based content accessible has inspired growing attention to improving visualization accessibility [26, 46]. In particular, projects have explored the use of screen readers to help BLV people interpret [25, 35, 51], navigate [57, 64], and even query visualizations [46, 49]. While these efforts have made valuable strides toward improving visualization accessibility, they have primarily considered scenarios involving a single chart. Another scenario involving visualizations that has been relatively underlooked, however, is that of *dashboards*.

Dashboards, commonly defined as a "*visual data representation structured as a tiled layout of simple charts and/or large numbers*" [44] typically include two or more coordinated visualizations of the same data, along with some way to filter that data and/or summarize it through key performance indicators (e.g., 'Total Sales' in the case of a business overview dashboard). Dashboards are often used to monitor and share data across domains (e.g., COVID reports, election results, summaries in business and finance applications). Compared to standalone charts, dashboards generally have a more complex structure in terms of both the cardinality and the types of *components* they use (e.g., charts, text, filtering widgets) [5, 17, 44, 62]. Furthermore, dashboards are typically highly *interactive*, enabling users to leverage the presented charts or widgets to filter the underlying data and analyze it from different perspectives [5, 44].

While the presence of multiple components and interactivity between those components make dashboards powerful interfaces for data exploration and analysis, a lack of design consideration for

[1] We use both *people first language* (people who are BLV) and *identity first language* (BLV people) depending on the grammar of a sentence, and in recognition that some people want their visual impairment acknowledged as an essential identifier and others do not. We also use the terms 'BLV people' and 'BLV users' interchangeably.

screen reader users (SRUs[2]) such as blind and visually impaired people often results in dashboards being inaccessible. For instance, in a study focusing on COVID-19 websites and dashboards, Siu et al. [52] found that most dashboards lacked even basic accessibility features such as alt-text and raw tabular data to complement charts. They further note that participant strategies to workaround this lack of access generally involved looking for alternative data sources (e.g., other websites, podcasts, news videos), switching consumption modalities (e.g., learning to use data sonification software), or even asking others to describe the data. Besides being a non-inclusive user experience, such strategies also prevent SRUs from extracting potential benefits from a dashboard that stem from the use of multiple (often complementary) visualizations and interactive data querying features of a dashboard.

Designing more accessible dashboards that provide a seamless dashboard usage experience for SRUs, however, requires answering a number of open questions. For instance, what should a dashboard's alt-text or description focus on? Should it emphasize just the charts and takeaways from those charts? Or should the description also summarize the composition and visual layout? How long should the description be and how should it be structured? How can dashboards best support navigation both across and within elements? Should elements be semantically grouped for navigation or should they preserve their visual order? How can SRUs be made aware of the interactive linkages between dashboard elements? How should a system summarize changes to the dashboard in response to an interaction? Are there certain changes that are more important to highlight than others?

Investigating such questions, we explore the design of web-based dashboards that support interaction via screen readers. Specifically, through a co-design process with two blind individuals (who are now also co-authors of this paper), we identify design goals for SRU-friendly dashboards pertaining to the dashboards' description, structure and navigation, and summarizing data changes. We operationalize these findings through a series of dashboard prototypes and a set of six (minimal) design goals for designing accessible visualization dashboards. We also encapsulate the implemented features into a reusable API for other developers to author accessible visualization dashboards. We evaluate the usability and utility of the generated dashboards through a study where five BLV users interacted with the dashboards to perform both targeted- and open-ended data analysis. We find that the generated dashboards afford flexible navigation and interaction both within and across components. Participant feedback also indicates that descriptions help them to understand the dashboard structure and promote interactive querying. Distilling findings from the co-design process and the user study, we discuss potential curb-cut effects for visualization systems and future research directions around supporting accessible and interactive data analysis, exploring alternative modalities, and improving dashboard accessibility at scale.

To summarize, we make the following contributions:

- We identify a set of design goals for creating dashboards accessible to BLV users interacting via a screen reader.
- We present a prototype system, Azimuth, that converts JSON-based dashboard specifications into web-based dashboards optimized for screen readers along with complementary descriptions to support dashboard comprehension and interaction.
- Based on a study with five BLV users, we discuss how dashboards generated by Azimuth grant access to the data at multiple levels of granularity, empowering users to not only answer given questions but also ask new questions of their own.

2 RELATED WORK

With the growing adoption of data visualizations across disciplines, making visualizations accessible to people with disabilities has become more important now than ever. Researchers have posed calls-to-action for investigating the design of inclusive visualizations to eliminate equity issues caused by a lack of access to data visualizations [26, 30, 36]. One idea among these provocations is to design accessible experiences for interacting with dashboards and multiple coordinated views. Kim et al. [26], for instance, survey 56 papers that focus on design of accessible visualizations through the use of non-visual modalities and highlight open questions around screen-reader based interaction design in the context of multiple coordinated views (e.g., how to enable cross-view filtering and brushing-and-linking?). Based on studies with SRUs accessing COVID-19 websites and dashboards, Sui et al. [52] and Fan et al. [19] recommend using multiple visualizations to fulfill users' diverse preferences for data abstraction and analysis. They note, however, that with existing technologies SRUs need to retain multiple pieces of information in their working memory while navigating between views, and call for more tightly coordinated views and interactions for data querying.

Given the relative lack of accessibility research on dashboards containing multiple connected visualizations, the remainder of this section focuses on summarizing the state of the art in making individual, stand alone visualizations accessible. That body of work primarily takes two approaches: text descriptions that summarize visualizations, and responsive visualizations that a user interacts with. We describe each in the next three paragraphs.

Text Descriptions for Visualizations. A series of research efforts (e.g., [2, 4, 22, 25, 35]) have investigated alt-text and descriptions for visualizations, providing design guidelines for the type of content to include. For instance, curating findings from studies on STEM education involving charts [6, 59], Ault et al. [4] present description generation guidelines for line charts including stating the chart type, labels, range of axis values, and key data patterns. Gould et al. [22] generalize such findings to charts images in general, and suggest keeping descriptions brief, clear and data-focused, and structured following an overview-then-details pattern [50]. Recent studies such as those by Jung et al. [25] and Lundguard and Satyanarayan [35] also echo similar findings, particularly stressing the importance of including objective statements focusing on data patterns over statements describing visual elements in a chart. However, none of these investigations answer the question of how multiple charts that display related data, or interactive features of

[2] We use the acronym SRU for screen reader users since screen reader use is not confined only to blind and visually impaired people

charts for filtering, both commonly found in dashboards, should be described.

Interactive Visualizations. Zong et al. [64] discuss design dimensions for expressive access to visualizations including structure, navigation, and descriptions and present a prototype that enables exploring a visualization at different levels of detail through multiple navigation strategies. The findings from the original study and prototype were also recently incorporated into a library that converts visualizations into a keyboard-navigable structure accessible to screen readers [7]. Chart Reader [57] takes a dataset and a chart specification as input, and generates structured line and bar charts that support a multi-level navigation experience. In particular, the resulting chart allows SRUs to walk-through a set of predefined insights, explore the chart axes or individual data points, and even leverage sonification to compare data series. HighCharts.js [23] is a JavaScript chart rendering library that provides built-in support for accessibility. Specifically, visualizations rendered with HighCharts are inherently structured to support navigating the different elements (axes, legends, marks) with a keyboard, include an auto-generated alt-text, and provide a default option to view the raw data table. Besides these default features, the library also provides additional APIs to configure support data series- and point-level sonification within a chart. Going beyond chart navigation, Sharif et al. [48, 49] introduced a JavaScript plug-in, VoxLens, that enables SRUs to not only get a summary and a sonified version of a chart but also supports keyword-based voice commands to drill-down and dynamically query for statistical information such as the min/-max, mean, etc. An evaluation of VoxLens for performing common analytic tasks showed that giving people this ability to explore and interactively query charts improved the accuracy of information extraction and interaction time by 122% and 36%, respectively. In recent follow-up work, Sharif et al. [47] extended VoxLens to support more drill-down operations including comparing data items and factor level categorization. These new additions furthered their prior findings and enabled screen-reader users to extract information with similar accuracy as non-screen-reader users. While the aforementioned systems help explore and interact with individual charts, they do not consider scenarios involving multiple coordinated views or cases where a web page element outside the chart such as a dropdown menu or a radio button updates/filters the data for a chart. In our work, we focus the context of dashboards where such scenarios are commonplace. In doing so, we explore support for navigation across components of a dashboard (as opposed to only within a single chart) as well as approaches to dynamically notify users about data changes in visualizations.

To summarize, there is a robust body of work exploring how to summarize, and interact with, individual charts. However, despite the general prevalence of dashboards across data domains [18, 20, 44, 62], there is a relative lack of work that provides guidance for how to make dashboards accessible—a gap we address through our research.

3 DESIGN PROCESS AND GOALS

We followed a participatory design process to ensure that our motivations were grounded in real challenges faced by SRUs and that any generated artifacts were usable and practically applicable [12, 21, 28]. Specifically, we collaborated with two blind individuals, Tim Harshbarger and Darrell Hilliker (the second and third author of this manuscript). Both individuals were initially recruited as participants because they are SRUs who frequently engage with websites and applications that involve data and visualizations, and have a collective experience of over 50 years in testing and designing accessibility features in software products. Their prior interactions with dashboards were primarily in an organizational setting predominantly for data monitoring and decision-making (similar to standard dashboard uses described by Sarikaya et al. [44]), and involved answering questions like "What type of defects are we running into?," "How many defects?," "How are the defects compared to the last quarter?," among others. Both authors also had personal experiences with dashboards on topics such as COVID-19 or elections. However, because fewer solutions were available to overcome accessibility barriers, these experiences were more limited. From the start, both Tim and Darrell were encouraged to consider becoming authors of any resulting publication if that was relevant and beneficial to them, and we mutually made the decision to make them authors due to the depth and importance of their contributions during our design process, which we describe next.

The design process lasted over four months, and included weekly or bi-weekly meetings for 30-60 minutes via video conferencing software. All sessions included one or more web-based dashboard prototypes that were used for discussion and were iteratively refined based on the feedback. During each session, either or both Tim and Darrell would share their screen and walk-through the prototypes, providing feedback on the implemented features. All sessions were recorded and the recordings along with session notes were used to fix issues and create alternative versions of the prototypes.

The prototypes were modeled on dashboards found on public visualization repositories such as Tableau Public [55] and Microsoft Power BI Data Stories Gallery [37], and covered a range of data domains including demographic survey results, healthcare, sporting events, flight delays, college admissions, and business data. Furthermore, the prototypes included both static (i.e., no two components were interactively linked) and interactive dashboards (i.e., interacting with one or more components would update another component), and the dashboards varied in complexity w.r.t. both the number (3–12) and type of components (e.g., charts, charts + widgets, charts + widgets + key performance indicators).

We implemented our dashboard prototypes on top of HighCharts.js [23] with its default accessibility features to render visualizations in the prototypes. This helped to ensure that the individual charts were accessible, enabling us to focus on exploring issues spanning beyond prior work on standalone charts (e.g., [25, 46, 64]) and instead focus the holistic experience of using a dashboard.

Combining the feedback from these design sessions with a review of prior work on SRU interaction with visualizations (e.g., [22, 25, 46, 52, 64]), we iteratively curated a list of six goals for designing accessible visualization dashboards. These goals correspond to the dashboard's structure (**DG1**, **DG2**), its textual description (**DG3**, **DG4**), and summarizing changes (**DG5**, **DG6**). Note that while these goals can serve as an initial set of general guidelines for authoring accessible dashboards, they are not an exhaustive list and were primarily curated to inform our dashboard prototypes detailed in the subsequent section.

DG1. The dashboard should use a consistent structure and create content hierarchies to aid navigation. Both prior work [52] and our design sessions highlighted that navigating dashboards is tedious and challenging due to the cardinality, variety (e.g., charts, KPIs, widgets), and unpredictable order of components. Following discussions from the co-design phase, we noted that an ideal dashboard should be designed similar to an accessible website in that it should use a consistent document object model (DOM) structure to render individual components and leverage appropriate HTML tags (e.g., headers and links) to facilitate flexible navigation strategies that allow sequentially perusing the dashboard or jumping to specific components ad-hoc.

DG2. The dashboard should present explicit filtering widgets for interactive charts and place all filters in a separate region. Filtering widgets (e.g., dropdown menus, checkboxes) enable interactive data querying within dashboards. Besides filtering widgets, however, visualization dashboards also tend to use view coordination techniques like cross-filtering or brushing-and-linking that allow selecting a mark in one chart to filter data in others [42]. While these are powerful interaction techniques for visual analysis, they are not commonly used outside visualization applications and may be unfamiliar to SRUs (both our blind co-authors had not realized that several dashboards they had encountered supported interactive querying). Thus, following our discussions, we noted that to make interactive querying capabilities of a dashboard both discoverable and accessible, any implicit mark-based filters should also be provided via explicit filtering widgets. Furthermore, during our design iterations, we also noted that the filters were easiest to use when grouped together and placed separately from other dashboard components as this helped preserve the dashboard layout and afforded quick navigation between filters.

DG3. Descriptions should summarize the dashboard's structure and interactivity, and highlight key data facts. Complementing prior work on alt text and descriptions for individual charts [2, 25, 35], we explored what descriptions should look like at a dashboard-level. In line with prior research on describing visual semantics of user interfaces for BLV people [39], our design sessions revealed that SRUs would benefit from a summary of the visual structure of a dashboard and the interactive relationships between dashboard components. In particular, we noted that such summaries could help orient new users and aid in creating a mental model of the dashboard, while also serving as a reference manual for returning users. Prior research has also shown that SRUs appreciate having a textual summary of key metrics and takeaways when working with dashboards [52]. Confirming this finding through our design sessions, we noted that dashboards should include automatically generated takeaways or "data facts" [53] as part of their description.

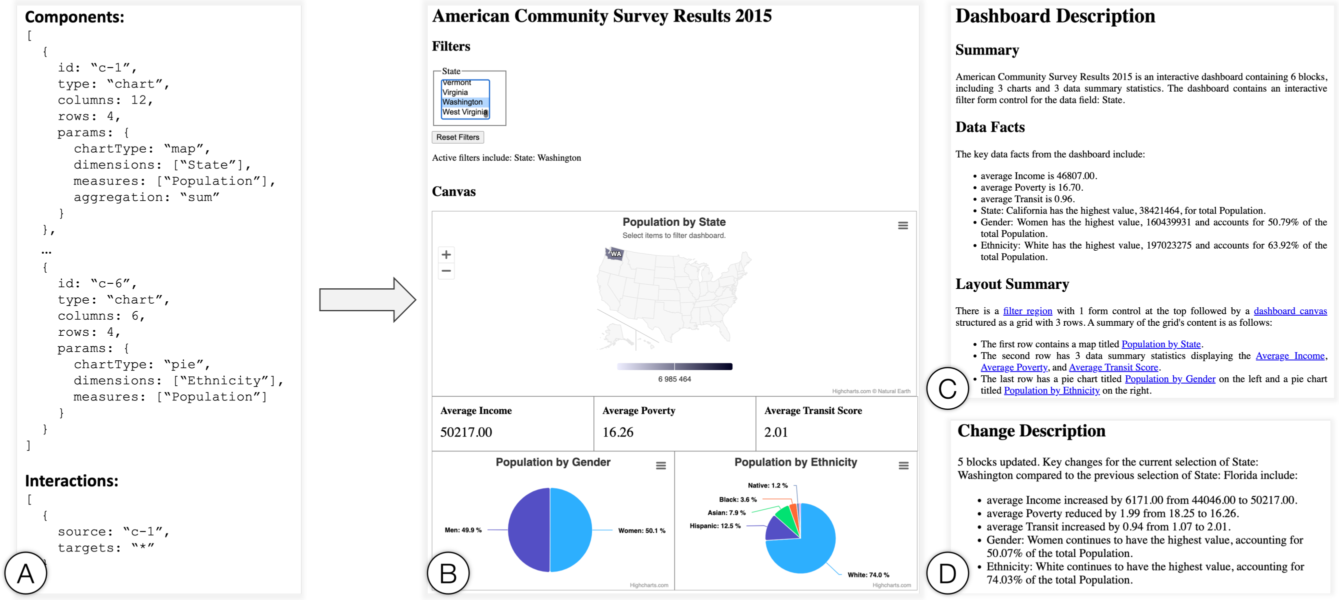
DG4. The dashboard description's structure should be optimized for screen readers. Given both the amount of information and the semantically different types of information covered in the description (visual and interaction summaries, key data facts), we realized that the structure of a dashboard's description was as important as the description's content. We observed that descriptions structured as paragraphs of plain text were difficult to follow and made it challenging to skip or jump to specific portions of the description. Thus, based on our design iterations and discussion, we note that an ideal dashboard description should leverage HTML tags like headers to segregate the description into logical chunks, use bulleted lists to facilitate easier reading, and also link references in the description to the individual components to aid dashboard navigation.

DG5. The dashboard should summarize key changes after data updates. Unlike static visualizations that persistently show the same data, data displayed in dashboards may update either via interactive filters or due to changes in the underlying data in monitoring scenarios. However, tracking changes after a data update can be challenging, requiring users to traverse the dashboard and mentally note changes compared to a previous state. Through our design studies, we noted that an explicit, dynamically generated description that exclusively summarizes data changes—referred to as a *change description*, can help overcome this challenge. Specifically, we explored different ideas for change descriptions including summarizing the visual changes to the dashboard, listing changes within individual components, listing key data facts for the updated dashboard, among others. Based on the iterative design and feedback process, we ultimately noted that a change description that summarizes both the visual extent of a change (i.e., how many components were affected) and any changes to the key data facts/-takeaways is most succinct and useful.

DG6. Change descriptions should support different modes of analytic inquiry. Prior work on accessibility in the context of general web design [8, 29] as well as data-focused websites [52] have shown that SRUs have diverse tasks and goals and merely providing access to the data and content is not enough. Along these lines, our design sessions revealed two types of tasks that users may perform via filters: 1) *subset analysis*, where the goal is to learn more about a specific data subset (selected via filters) and 2) *comparison*, where the goal is to compare two data subsets. To this end, we note that besides including change descriptions following **DG5**, ideal dashboards should also provide users the option to adjust the change description to fit their analytic goals.

4 AZIMUTH

Incorporating the aforementioned design goals, we implemented Azimuth as a prototype system to generate dashboards optimized for SRUs. Figure 2 presents an overview of the system architecture and Figures 1 & 3 show examples of dashboard generated by Azimuth. Specifically, given a dataset and a dashboard specification (Figure 1A), the system parses the specification to generate a web page with a dashboard inherently structured for smooth navigation via screen readers (Figure 1B, **DG1-2**). Besides the dashboard itself, Azimuth also generates a textual description to summarize the dashboard and its key takeaways (Figure 1C, **DG3-4**). Finally, as users interact with the generated dashboard and apply filters, the system dynamically populates and updates summaries of the data changes through a textual change description (Figure 1D, **DG5-6**).



Components:

```
[
  {
    id: "c-1",
    type: "chart",
    columns: 12,
    rows: 4,
    params: {
      chartType: "map",
      dimensions: ["State"],
      measures: ["Population"],
      aggregation: "sum"
    }
  },
  …
  {
    id: "c-6",
    type: "chart",
    columns: 6,
    rows: 4,
    params: {
      chartType: "pie",
      dimensions: ["Ethnicity"],
      measures: ["Population"]
    }
  }
]
```

Interactions:

```
[
  {
    source: "c-1",
    targets: "*"
  }
```

American Community Survey Results 2015

Filters

State

Vermont
Virginia
Washington
West Virginia

Reset Filters

Active filters include: State: Washington

Canvas

Average Income	Average Poverty	Average Transit Score
50217.00	16.26	2.01

Dashboard Description

Summary

American Community Survey Results 2015 is an interactive dashboard containing 6 blocks, including 3 charts and 3 data summary statistics. The dashboard contains an interactive filter form control for the data field: State.

Data Facts

The key data facts from the dashboard include:

- average Income is 46807.00.
- average Poverty is 16.70.
- average Transit is 0.96.
- State: California has the highest value, 38421464, for total Population.
- Gender: Women has the highest value, 160439931 and accounts for 50.79% of the total Population.
- Ethnicity: White has the highest value, 197023275 and accounts for 63.92% of the total Population.

Layout Summary

There is a filter region with 1 form control at the top followed by a dashboard canvas structured as a grid with 3 rows. A summary of the grid's content is as follows:

- The first row contains a map titled Population by State.
- The second row has 3 data summary statistics displaying the Average Income, Average Poverty, and Average Transit Score.
- The last row has a pie chart titled Population by Gender on the left and a pie chart titled Population by Ethnicity on the right.

Change Description

5 blocks updated. Key changes for the current selection of State: Washington compared to the previous selection of State: Florida include:

- average Income increased by 6171.00 from 44046.00 to 50217.00.
- average Poverty reduced by 1.99 from 18.25 to 16.26.
- average Transit increased by 0.94 from 1.07 to 2.01.
- Gender: Women continues to have the highest value, accounting for 50.07% of the total Population.
- Ethnicity: White continues to have the highest value, accounting for 74.03% of the total Population.

Figure 1: Example of Azimuth's input and output. Given a dashboard specification (A), the system generates a web page containing a dashboard (B), a textual description of the dashboard (C), and dynamically populates a change description (D) to summarize data changes in the dashboard resulting from user interactions such as filtering.

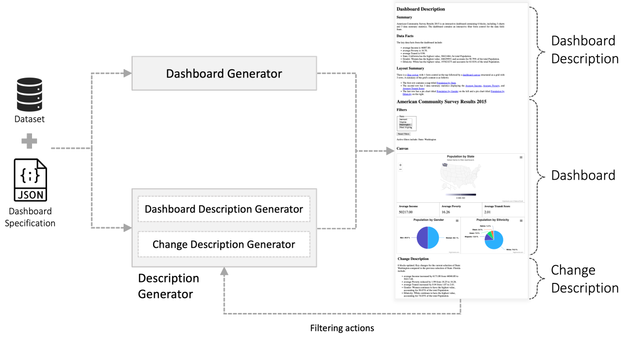


Figure 2: System architecture overview.

This combination of the structured dashboard and the complementary descriptions allows SRUs to gain an overview using the description when they first open the web page and then dive into the dashboard canvas for analysis. Once users know of a dashboard, however, they can also jump straight to the canvas or the change description to identify key data changes.

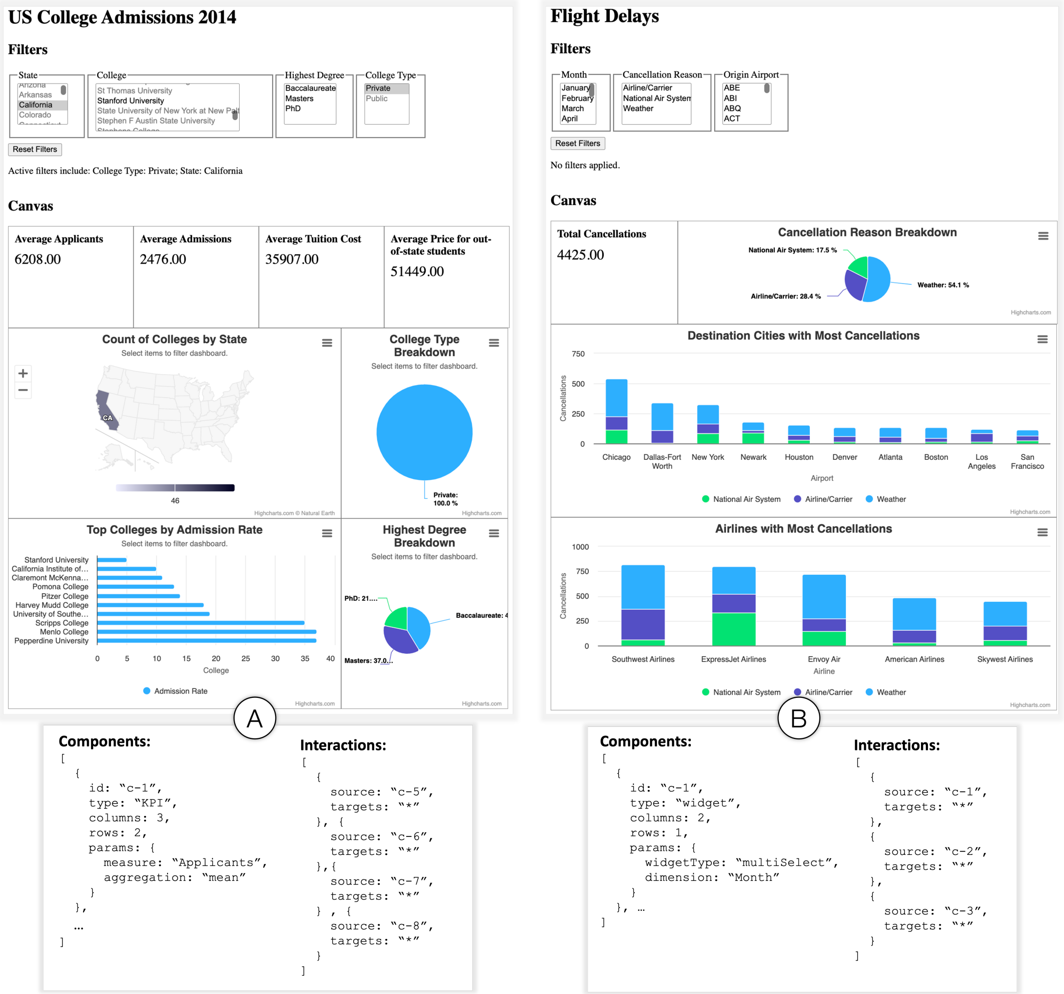
We opted for a specification-driven approach to rapidly prototype examples during our design phase and also make it possible for other developers to specify similar dashboards. For chart components in particular, we used an abstract specification (including `chartType`, `dimensions`, `measures`) to be agnostic of rendering libraries and try different alternatives during prototyping (e.g., D3.js [10], Visa chart components [60], Highcharts.js [23]). However, future implementations can replace this abstract specification with more widely used visualization grammars like Vega-Lite [45] that support more granular specification of charts. Our current implementation of Azimuth uses JSON for dashboard specification; The dashboard (Figure 1B) is rendered as standard web page content using HTML, CSS, and JavaScript. Specifically for rendering visualizations, we use the HighCharts library [23] as it supports built-in accessibility features like viewing the raw data and performing mark-level navigation at an individual chart level.

4.1 Dashboard Specification

The input specification (Figure 1A) is a JSON object structured as a list of dashboard components and a list of interactions between those components [17]. Currently supported component types include charts, key performance indicators, widgets, and text blocks.

Chart Types. The current prototype supports bar charts (including single, stacked, grouped, and diverging bar charts), pie charts, histograms, scatterplots, line charts (including single and multi-series charts), and choropleth maps. Charts visualize one or more quantitative attributes (`measures`) and optionally break down the values across other categorical, geographic, or temporal attributes (`dimensions`). The input specification provides fields to specify the chart type, measures, and dimensions, along with optional fields to specify a chart title and aggregation for measures (e.g., sum, average).

Key performance indicators (KPI components). KPI components are glanceable numbers that summarize a single measure (e.g., ‘Total Population,’ ‘Average Income’). The input specification for the KPI component provides parameters to specify a `measure` and an `aggregation` that are used to dynamically compute the KPI value



US College Admissions 2014

Filters

State: Arizona, Arkansas, California (selected), Colorado, Connecticut
College: St Thomas University, Stanford University, State University of New York at New Palt, Stephen F Austin State University, Stephens College
Highest Degree: Baccalaureate, Masters, PhD
College Type: Private (selected), Public

Reset Filters

Active filters include: College Type: Private; State: California

Canvas

Average Applicants	Average Admissions	Average Tuition Cost	Average Price for out-of-state students
6208.00	2476.00	35907.00	51449.00

Flight Delays

Filters

Month: January, February, March, April
Cancellation Reason: Airline/Carrier, National Air System, Weather
Origin Airport: ABE, ABI, ABQ, ACT

Reset Filters

No filters applied.

Canvas

Total Cancellations
4425.00

A

Components:
```
[
  {
    id: "c-1",
    type: "KPI",
    columns: 3,
    rows: 2,
    params: {
      measure: "Applicants",
      aggregation: "mean"
    }
  },
  …
]
```

Interactions:
```
[
  {
    source: "c-5",
    targets: "*"
  }, {
    source: "c-6",
    targets: "*"
  },{
    source: "c-7",
    targets: "*"
  } , {
    source: "c-8",
    targets: "*"
  }
]
```

B

Components:
```
[
  {
    id: "c-1",
    type: "widget",
    columns: 2,
    rows: 1,
    params: {
      widgetType: "multiSelect",
      dimension: "Month"
    }
  }, …
]
```

Interactions:
```
[
  {
    source: "c-1",
    targets: "*"
  },
  {
    source: "c-2",
    targets: "*"
  },
  {
    source: "c-3",
    targets: "*"
  }
]
```

Figure 3: Example dashboards generated by AZIMUTH. (A) A dashboard showing US college admission data. Here, the four charts are interactive and can be implicitly used to filter the dashboard data. Correspondingly, AZIMUTH generates four filter widgets using the underlying dimensions in the charts to make this filtering capability explicit. (B) A dashboard summarizing flight delays in US airports. Here, the three filter widgets (Month, Cancellation Reason, and Origin Airport) are explicitly defined in the input specification and the charts themselves are static (i.e., marks on a chart cannot be used to filter the dashboard).

when rendering the dashboard. An example of a KPI component specification is shown in Figure 3A-bottom.

Widgets. AZIMUTH currently supports single- and multi-select dropdown menus that are commonly used in dashboards to support filtering using `dimension` values [5, 44, 62]. Widgets can be defined in the input specification by providing a `widgetType` (single vs. multi-select) and a `dimension` as parameters. An example of a widget component specification is shown in Figure 3B-bottom.

Layout. Component specification in AZIMUTH includes options for indicating the size of individual components in terms of number of `rows` and `columns` of a grid layout and for additional (optional) metadata fields such as `title`. The current implementation supports a responsive 12-column wide grid structure modeled after the popular website design framework, Bootstrap [1]. For example, in Figure 1A, defining the value of the `columns` field for the map component as 12 results in a map spanning the entire width of the dashboard, as shown in Figure 1B.

Interactions. A chart or widget component can be interactively linked to other chart and KPI components by specifying the `targets` key in the list of *interactions*. For instance, in Figure 1A, the "*" in the `targets` field for the map component (`c-1`) indicates that the map is used to filter the two pie charts and the three KPI components in the dashboard. As described in more depth in subsection 4.2, AZIMUTH will also automatically add explicit filter widgets if needed to improve the discoverability and use of filters.

4.2 Dashboard Structure and Navigation

To improve readability during navigation, our system ensures that each component has a meaningful title even if it is not explicitly defined in the input specification. For instance, the input specification in Figure 1 does not specify a `title` for the chart components. In such cases, Azimuth uses the `measures` and `dimensions` to inherently generate a basic title like '*Population by State*' or '*Population by Gender*' (Figure 1B).

Following the design goal of presenting explicit filtering widgets for interactive charts (**DG2**), the system detects interactive chart components in the input specification and adds a filter widget (e.g., dropdown menu, radio button) for the underlying dimension if a widget for that dimension is not already specified. An example of this is shown in Figure 1 where the map is used to implicitly filter all other components (i.e., clicking on states in the map updates the data in the dashboard). In this case, Azimuth detects that there is no explicit filtering widget for the *State* field and creates one, allowing users to apply the same filter using the widget or the marks (states) in the map. Similarly, the four filtering widgets in Figure 3A (for attributes *State*, *College*, *Highest Degree*, and *College Type*) are also generated by the system. Besides making filters explicit through widgets, for consistency, we also ensure that the widgets are dynamically synchronized with their chart counterparts. In other words, if applying a filter removes marks from a chart, the values corresponding to the removed marks should also be disabled in the widget. Figure 3A shows an example of this where a subset of values (e.g., St Thomas University, Stephen F Austin State University) in the *College* widget are disabled because they do not meet the active filtering criteria (*State*: California, *College Type*: Private).

In terms of the underlying DOM, the dashboard is broken down into a filter and a canvas region to make the presence of interactive filtering capabilities more apparent (**DG2**). Furthermore, per **DG1** and general web accessibility design practices, we employ a consistent heading structure (e.g., individual dashboard components are all at the same heading level) and use standardized HTML form controls like `<select>` (**DG1**). This consistent use of standardized web elements enables users to flexibly navigate the dashboard and leverage built-in screen reader shortcuts to jump to specific portions ad-hoc (e.g., using the '*F*' key in JAWS to cycle through filters).

4.3 Dashboard Description

Azimuth also generates a description that summarizes the dashboard's content and key data takeaways (**DG3**). For instance, InfoBox 1 presents the description generated for the dashboard in Figure 3B. Following **DG4**, the description is broken down into sections to facilitate easier reading and navigation. We elaborate on the individual description sections and their generation logic below.

Dashboard Summary. To give users a sense for the dashboard's content, the description first provides a brief alt text-like summary covering metadata including the dashboard title, the number and type of components, as well as if there are any interactive filtering options. This summary is generated using a templated string that is parameterized based on the input dashboard specification.

Summary
Flight Delays is an interactive dashboard containing 4 blocks, including 3 charts and 1 data summary statistic. The dashboard contains 3 interactive filter form controls for the data fields: Month, Cancellation Reason, and Origin Airport.

Data Facts

- Total Cancellations is 4425.00.
- Cancellation Reason: Weather has the highest value, 2393 and accounts for 54.08% of the total Cancellations.
- City: Chicago has the highest value, 541, for total Cancellations and is composed of Weather: 58.04%, National Air System: 21.44%, Airline/Carrier: 20.52%.
- Airline: Southwest Airlines has the highest value, 818, for total Cancellations and is composed of Weather: 54.52%, Airline/Carrier: 37.78%, National Air System: 7.70%.

Layout Summary
There is a filter region with 3 form controls at the top followed by a dashboard canvas structured as a grid with 3 rows. A summary of the grid's content is as follows:

- The first row contains 1 data summary statistic displaying the Total Cancellations on the left and 1 pie chart titled Cancellation Reason Breakdown on the right.
- The second row contains a stacked bar chart titled Destination Cities with Most Cancellations.
- The last row contains a stacked bar chart titled Airlines with Most Cancellations.

InfoBox 1: System generated description for the flight delays dashboard in Figure 3B.

Data Facts. The description also lists the key data facts from the dashboard. These data facts include a call-out of any KPIs and one key takeaway from each chart in the dashboard. Specifically, Azimuth inspects the data underlying a chart and uses a set of statistical functions and heuristics to identify salient data values or trends. These heuristics are defined based on prior work on mappings between specific chart types and analytic tasks [11, 27, 41, 43], and systems that generate data fact- and insight-recommendations [14, 16, 53, 56, 61]. For example, for a bar chart, Azimuth generates data facts corresponding to categories with the min/max values, for scatterplots, the system generates facts corresponding to the correlation between the displayed fields, for pie charts, the system talks about the distribution of values, and so on.

In terms of the number of facts listed in the description, our design sessions revealed that one fact corresponding to each chart or KPI component was an optimal count. Furthermore, after discussing multiple potential ordering strategies for the data facts during the phase (e.g., following the visual layout order, largest charts first, a computed statistical "interestingness" metric), we finally settled on ordering the data facts following the visual sequence of the dashboard with one exception: any KPI components in the dashboard are called out first before any charts due their prevalence and general importance in dashboards [3, 62]. An example of this prioritized ordering is displayed in the description in Figure 1C where the three KPI values (Average Income, Average Poverty, Average Transit) are listed before the data fact corresponding to the map even though the map appears before the KPIs in the dashboard.

Layout Summary. Lastly, the help users create a mental model of the dashboard, Azimuth also summarizes the visual layout of the dashboard canvas as part of its description. The summary starts with an overview of the two main sections of the dashboard including the filters and the main canvas. To summarize the canvas layout, the system iterates through the dashboard grid following the natural reading order of left to right and top to bottom. Next, the system detects groups of rows or columns based on the component sizes and generates one bullet point summarizing each group. For instance, in Figure 3B, the system detects three rows based on the component heights and widths, and correspondingly generates one statement summarizing the contents of each row (InfoBox 1). When describing a row or column, the system lists the component types and titles as well as their relative size and positions. Finally, the layout summary also serves as an interactive index to the dashboard, linking the titles to the corresponding component headings in the main dashboard (underlined text in InfoBox 1).

8 blocks updated. For the selection of College Type: Private, key observations include:

- Average Tuition Cost is 29548.
- Average Price for out-of-state students is 43155.
- Average Applicants is 4558.
- Average Admissions is 2290.
- State: New York has the highest value, 74, for total Count.
- College: Stanford University has the lowest value, 5, for average Admission Rate.
- Highest Degree: Baccalaureate has the highest value, 432, accounting for 51.37% of the total Count.

8 blocks updated. For the selection of College Type: Public, key observations include:

- Average Tuition Cost is 8443.
- Average Price for out-of-state students is 32789.
- Average Applicants is 10043.
- Average Admissions is 6018.
- State: New York has the highest value, 34, for total Count.
- College: University of California Berkeley has the lowest value, 17, for average Admission Rate.
- Highest Degree: Masters has the highest value, 237, accounting for 49.79% of the total Count.

InfoBox 2: Examples of change descriptions generated in the 'Summary' mode for the college admissions dashboard in Figure 3A. (Left) Description when the filter is set to {*College Type: Private*}. (Right) Descriptions when the filter is set to {*College Type: Public*}.

8 blocks updated. Key changes for the current selection of College Type: Public compared to the previous selection of College Type: Private include:

- Average Tuition Cost reduced by 21105 from 29548 to 8443.
- Average Price for out-of-state students reduced by 10366 from 43155 to 32789.
- Average Applicants increased by 5485 from 4558 to 10043.
- Average Admissions increased by 3728 from 2290 to 6018.
- State: New York continues to have the highest value, 34, for total Count.
- College: University of California-Berkeley replaced Dartmouth College as the category with the lowest value, 17, for average Admission Rate.
- Highest Degree: Masters replaced Baccalaureate as the category with the highest value, accounting for 49.79% of the total Count.

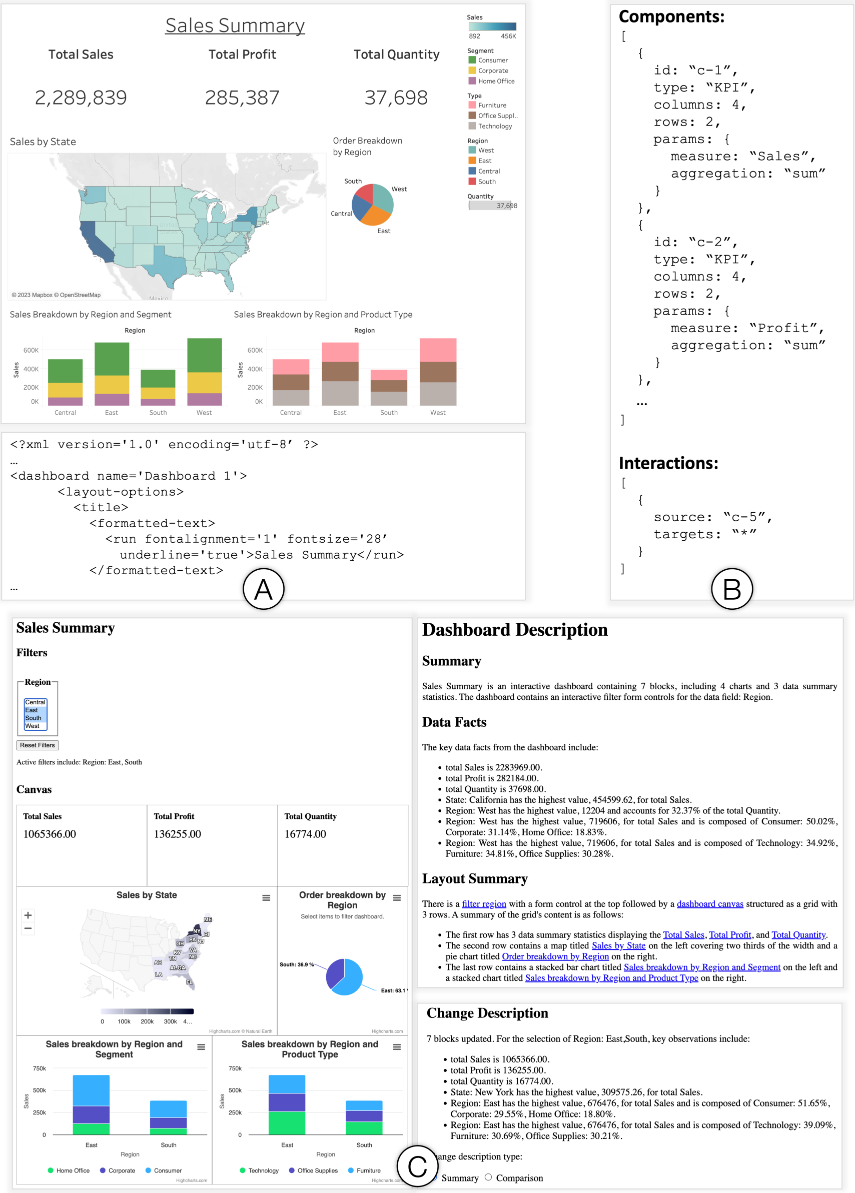
InfoBox 3: Example of a change description generated in the 'Comparison' mode for the same set of filters as in InfoBox 2 ({*College Type*: Private} followed by {*College Type*: Public}). In this case, instead of merely stating the values for the current dashboard state, the system contextualizes the values relative to the preceding state.

4.4 Change Description

As stated earlier, a key feature that dashboards offer is the ability to actively monitor real-time data or dynamically query data by applying different filters. To help SRUs comprehend and track changes between states of the dashboard, Azimuth also dynamically generates a textual "change description" (**DG5**). Specifically, the change description summarizes the number of dashboard components that were impacted by a data change, lists the active data filters, and lists a set of key data facts similar to those in the original dashboard description. To generate the data facts, the system uses the same logic as that used to generate facts in the dashboard description.

Within the DOM, the `aria-live` attribute for the change description is set to '*polite*' so the system automatically alerts users whenever the dashboard is updated. However, the description is also persisted on the page, allowing users to manually navigate the description if they want to revisit specific portions of the text.

Following **DG6**, Azimuth provides two modes for analytic inquiry via the change description—namely, *summary* and *comparison*. In the summary mode (default), the system lists the data facts for the active state of the dashboard, making it equivalent to the facts in the dashboard description but for a subset of the data. When in the comparison mode, the change description is phrased to not only state the key takeaways from the active state of the dashboard but also to relativize them to the previous state of the dashboard. InfoBox 2 and InfoBox 3 highlight differences in the two change description modes using the example of applying the filters {*College Type*: Private} followed by {*College Type*: Public} in the context of college admissions dashboard shown in Figure 3A. Specifically, in the case of InfoBox 2, both change descriptions use the same



```
<?xml version='1.0' encoding='utf-8' ?>
…
<dashboard name='Dashboard 1'>
      <layout-options>
        <title>
          <formatted-text>
            <run fontalignment='1' fontsize='28'
              underline='true'>Sales Summary</run>
          </formatted-text>
…
```

Components:

```
[
  {
    id: "c-1",
    type: "KPI",
    columns: 4,
    rows: 2,
    params: {
      measure: "Sales",
      aggregation: "sum"
    }
  },
  {
    id: "c-2",
    type: "KPI",
    columns: 4,
    rows: 2,
    params: {
      measure: "Profit",
      aggregation: "sum"
    }
  },
  …
]
```

Interactions:

```
[
  {
    source: "c-5",
    targets: "*"
  }
]
```

Sales Summary

Filters

Region

Central
East
South
West

Reset Filters

Active filters include: Region: East, South

Canvas

Total Sales	Total Profit	Total Quantity
1065366.00	136255.00	16774.00

Dashboard Description

Summary

Sales Summary is an interactive dashboard containing 7 blocks, including 4 charts and 3 data summary statistics. The dashboard contains an interactive filter form controls for the data field: Region.

Data Facts

The key data facts from the dashboard include:

- total Sales is 2283969.00.
- total Profit is 282184.00.
- total Quantity is 37698.00.
- State: California has the highest value, 454599.62, for total Sales.
- Region: West has the highest value, 12204 and accounts for 32.37% of the total Quantity.
- Region: West has the highest value, 719606, for total Sales and is composed of Consumer: 50.02%, Corporate: 31.14%, Home Office: 18.83%.
- Region: West has the highest value, 719606, for total Sales and is composed of Technology: 34.92%, Furniture: 34.81%, Office Supplies: 30.28%.

Layout Summary

There is a filter region with a form control at the top followed by a dashboard canvas structured as a grid with 3 rows. A summary of the grid's content is as follows:

- The first row has 3 data summary statistics displaying the Total Sales, Total Profit, and Total Quantity.
- The second row contains a map titled Sales by State on the left covering two thirds of the width and a pie chart titled Order breakdown by Region on the right.
- The last row contains a stacked bar chart titled Sales breakdown by Region and Segment on the left and a stacked chart titled Sales breakdown by Region and Product Type on the right.

Change Description

7 blocks updated. For the selection of Region: East,South, key observations include:

- total Sales is 1065366.00.
- total Profit is 136255.00.
- total Quantity is 16774.00.
- State: New York has the highest value, 309575.26, for total Sales.
- Region: East has the highest value, 676476, for total Sales and is composed of Consumer: 51.65%, Corporate: 29.55%, Home Office: 18.80%.
- Region: East has the highest value, 676476, for total Sales and is composed of Technology: 39.09%, Furniture: 30.69%, Office Supplies: 30.21%.

Change description type:

● Summary ○ Comparison

Figure 4: Example of using AZIMUTH to recreate a Tableau dashboard. Given a Tableau dashboard with its underlying XML specification (A), we parse the XML to generate a dashboard specification in the format AZIMUTH supports (B), and use this specification to generate a new web page with descriptions and an interactive dashboard (C).

phrasing but focus on Private (left) and Public (right) colleges, respectively. In InfoBox 3, however, when the user switches to {*College Type*: Public}, the system not only summarizes the key facts but also how they differ relative to the preceding selection of {*College Type*: Private}.

To further enhance the interactive querying experience, based on our testing and the frequency of specific operations during the design phase we also provide three shortcut key combinations as part of the generated web page. In particular, pressing `Alt + X` resets any active filters, allowing users to restore the dashboard to its default state. Pressing *Alt + - (minus)* at any time reads the contents of the change description without users having to navigate to that section of the page. Lastly, `Alt + M` toggles the change description verbosity between 'Summary' and 'Comparison.' Note that while these shortcuts are designed to support a more fluid and faster user experience, AZIMUTH still provides explicit HTML elements (a button to reset filters, a header to jump to the change description section, and radio buttons to toggle the change description mode) for users who do not prefer or are unfamiliar with the shortcut keys.

4.5 API and Applications

Azimuth is implemented in JavaScript and is structured as an API to help visualization developers create dashboards similar to those shown in Figures 1 & 3. Below we discuss two examples illustrating how Azimuth can help developers replace their dashboards with a dashboard including all the features showcased above, or only use a portion of Azimuth's output to enhance an existing dashboard rendered using a different library.

Recreating Tableau Dashboards. As a mainstream visualization tool, Tableau is one of the most popular tools for authoring dashboards. While there exist suggestions to design accessible content with Tableau [54], authors do not always incorporate these suggestions, often resulting in inaccessible dashboards. To circumvent accessibility issues in such cases, Azimuth can be used to recreate an accessible alternative of a Tableau dashboard (Figure 4). As an example, we implemented a web application that converts the XML specification underlying Tableau dashboards into Azimuth's input specification and uses this specification to create an alternative dashboard (Listing 1).

From a development standpoint, after converting the Tableau XML into Azimuth's input specification, incorporating this functionality is a two step process involving: 1) initializing Azimuth with the dataset and dashboard specification (Listing 1, line 3) and 2) invoking the functions to render the dashboard and descriptions in their intended DOM elements (Listing 1, lines 4-6). Azimuth internally uses its pre-defined templates and the built-in statistics module to generate the descriptions, Highcharts.js [23] and HTML/CSS to render the dashboard contents, and also defines the required event handing functions to apply any specified interactions. For instance, the Tableau dashboard in Figure 4A is configured such that interacting with the pie chart filters the rest of the dashboard. Detecting this from the input specification, Azimuth implements a similarly interactive pie chart but per **DG2**, also creates an explicit filter widget for the Region attribute (the `dimension` displayed in the pie chart).

Summary

Coronavirus Spreads is an interactive dashboard containing 2 chart blocks. The dashboard contains 2 interactive filters for the data fields: Date and Country/Region.

Data Facts

- Country/Region: US has the highest value, 33264 for total New Cases on Date: 4/4/20.
- Country/Region: US has the highest value, 366613 for the total New Cases.

Layout Summary

The dashboard canvas is structured as a grid with 2 rows. A summary of the grid's content is as follows:

- The first row contains a scatterplot titled Total New Cases by Date and Country/Region.
- The second row contains a bar chart titled Total New Cases by Country/Region.

```
1  let dashboardSpec = getAzimuthSpec(twbFile); //
   ↪  custom function to convert the Tableau XML
   ↪  specification (.twb file) into Azimuth's
   ↪  dashboard specification format
2
3  azimuth.init(dashboardSpec, dataFile); // dataFile
   ↪  can be a CSV or JSON file
4  azimuth.generateDescription('#descriptionDiv');
5  azimuth.generateDashboard('#dashboardDiv');
6  azimuth.generateChangeDescription('#changeDescript⌟
   ↪  ionDiv');
```

Listing 1: JavaScript code to recreate a Tableau dashboard with Azimuth as shown in Figure 4.

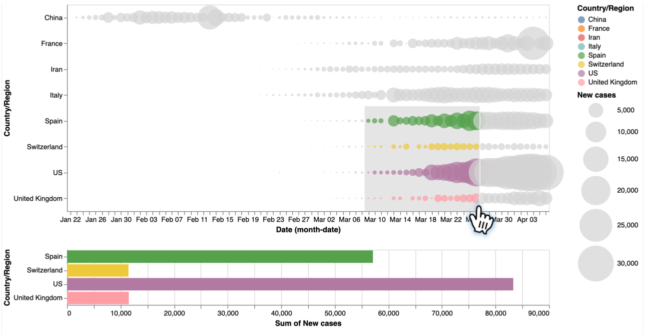


Figure 5: A multiple coordinated view display of COVID-19 data created using Vega-Lite [45]. In this case, brushing over items in the scatterplot filters the data shown in the bar chart. Azimuth's generated description for this display is shown in InfoBox 4

InfoBox 4: Azimuth's generated description for the Vega-Lite multiple coordinated view display in Figure 5.

Generating a description for a Vega-Lite dashboard. Instead of generating an entire web page, Azimuth can also be used to augment the accessibility of existing dashboards with textual descriptions. For instance, consider the multiple coordinated views in Figure 5[3] created with Vega-Lite [45]. If a developer wanted to preserve the Vega-Lite display in their application but leverage Azimuth only to generate the description, they can do so by converting the Vega-Lite specification into Azimuth's input specification, initializing the Azimuth API with this specification, and then making the function call `generateDescription('#descriptionDiv', false)`. Here, the first parameter is the target DOM element where the description is rendered and the second parameter is a boolean flag (`rendering`) indicating that Azimuth is not used to render the dashboard itself. This flag is used to adjust the description to not include details about any explicit filter controls or create hyperlinks to the dashboard elements since Azimuth has no control over how the original

[3] Source for chart: https://towardsdatascience.com/interactive-data-visualization-for-exploring-coronavirus-spreads-f33cabc64043

dashboard is rendered. The resulting description generated by Azimuth is presented in InfoBox 4. Notice that although there are no hyperlinks or filter widget references as in the case of descriptions accompanying an Azimuth dashboard (e.g., InfoBox 1), the summary statement still includes information about interactivity and the data fields that can be used for interactive querying.

5 PRELIMINARY EVALUATION

We used the prototyped dashboards as a test bed for a user study to assess the usability and utility of the proposed dashboard interface. Specifically, we wanted to collect feedback on: 1) the three core components of the generated dashboards (dashboard description, dashboard structure, change description) and 2) the generated dashboards' support for targeted and open-ended data analysis.

5.1 Participants and Setup

We recruited five BLV participants (four male, one female) for a 90-minute study by sharing a call for participation through our collaborators in the blind community. We ended recruitment when we felt we had reached saturation, which was facilitated by the fact that most participant comments reflected early findings from our co-design session. Four participants were totally blind and one was almost totally blind with some light perception. All participants were based in the US and were spread across four states. Sessions were conducted remotely via Zoom and were recorded with permission from the participants. Participants were provided with links to the prototypes and study tasks, and completed the study on their own computers while sharing their screens. Participants were offered a compensation of a $75 Amazon Gift Card. Four participants accepted this compensation format and one requested that the amount be donated to a charitable organization of our choice[4].

Four participants performed the study on a Windows OS with two of those participants using the JAWS screen reader and two using NVDA. One participant used a Mac OS with VoiceOver as the screen reader. All participants had previously interacted web-based visualizations with three participants self-reporting that they encountered visualizations on a daily/weekly basis and the other two on a monthly/quarterly basis. All but one of the participants were aware of dashboards as a concept. Of these four participants, two participants had casually interacted with one-off dashboards in the past whereas the other two participants had worked with dashboards on multiple occasions in a professional setting. Both participants with prior experience with dashboards said they had either ended up working with the raw data in Excel or sought external help to infer information because the dashboards were inaccessible.

5.2 Procedure

The study protocol, dashboards, and tasks were finalized based on discussions during the co-design process. Each study session was broken down into four parts:

Introduction and Training [~25min]. Sessions began with participants answering a series of demographic and background questions detailing their experience with data, visualizations, and dashboards. This was followed by an introduction to the our proposed dashboard interface using a dashboard displaying the American Community Survey results (Figure 1). Specifically, participants were asked to first go through the dashboard description and subsequently the main dashboard and change descriptions until they felt comfortable with the different components.

Task Phase 1: Targeted Analysis [~30min]. This phase was designed to emulate a scenario of a SRU working with a dashboard to answer specific questions (e.g., answering questions about defects as discussed in the formative study) and implicitly assess how well the developed prototypes supported such scenarios. Participants were given a dashboard about Flight Delays in the US (Figure 3B) and were provided with a list of eight tasks/questions to answer using the dashboard. The tasks were designed to test different features of the interface and included a mix of questions that required participants to navigate within individual components (e.g., *Is ExpressJet one of the top three airlines in terms of total number of cancellations?*), perform filtering and comparisons using the change description (e.g., *For the SEA airport, which airline had the most cancellations due to Airline/Carrier?*, *Between November and December, which month had more flight cancellations?*), or some combination of filtering and inspecting individual components (e.g., *Which three destination cities have had the most cancellations for flights originating from the ATL airport?*). The tasks were provided through a text file and their order was randomized across sessions. We asked participants to refresh the page before each task to have a common starting state for the dashboard. Participants were encouraged to attempt as many tasks as possible within an approximate duration of 30 minutes. They were not instructed to optimize for speed on individual tasks, because our goal was not to assess completion time, but rather to get general feedback on the interface and interactions. To this end, participants were also permitted (and even encouraged) to provide feedback and discuss specific features amidst a task.

Task Phase 2: Open-ended Exploration [~20min]. Complementing the targeted analysis scenario, this second task phase emulated a scenario of encountering a publicly available dashboard on the web and exploring the data in an open-ended manner. Specifically, participants were given a dashboard about US College admissions (Figure 3A) and were asked to freely explore the dashboard and investigate any questions that came to mind. If they were trying to answer specific questions, we requested participants to speak the question aloud so we could observe their approach to answering the question.

Debrief [~15min]. Lastly, we conducted a semi-structured interview where participants were asked a series of questions about their experience with the prototypes. In cases where participants had used dashboards before, we also inquired about their experience relative to other dashboards.

5.3 Analysis

The studies resulted in approximately 7.5 hours of session recordings including both screen capture and audio. In line with the two study goals, we manually inspected the screen recordings and transcripts to capture feedback on the core components of the generated dashboards and the supported analysis scenarios, respectively. We

[4] We donated the $75 to The Lighthouse for the Blind, Inc., a private, not-for-profit social enterprise providing employment, support, and training opportunities for people who are blind, DeafBlind, and blind with other disabilities.

also followed an iterative affinity diagramming process grouping statements from the transcript to identify other general themes that organically arose during the debriefing phase. Below we discuss these themes and the results of the analysis.

5.4 Results and Discussion

Dashboard descriptions provide a comprehensive overview. All participants read through the descriptions not only in the training phase but also during the tasks phases where it was optional. We observed that participants particularly slowed down when reading the layout summary and created a mental image of the dashboard that they later leveraged for navigation. For instance, after going through the layout summary for the flights dashboard in Figure 3B, P2, in a confirmatory tone, asked "*Ok so there is one number and a pie chart next to it, and then two big bar charts below that, is that right?*" Furthermore, participants unanimously commented that the three sections within the description presented different but equally important information, validating the underlying design principles. P1, for example, said "*all of it was really useful because it basically gave me like an introduction as well as the main messages [data facts] and general items that I was going to see in there and how they were presented.*" Lastly, all participants found the description structure easy to navigate with P2 even commending the phrasing of individual statements and the careful use of punctuation marks to emphasize data attributes and values.

The dashboard structure supports flexible navigation. All participants were comfortable navigating the dashboard and individual charts, referring to the experience as "*seamless*" (P5) and "*smooth*" (P4). Interestingly, even within the five participants, we observed varied navigation strategies that participants employed. P4, for example, often used the 'H' key with JAWS to navigate back and forth through all headings until he reached a dashboard component that he was interested in diving further into. On the other hand, P5 would first spend some time going through the list of headers to vaguely memorize them and then used VoiceOver's item chooser feature to rapidly jump to specific headings or form controls. Appreciating how the dashboard structure allowed her to leverage her existing knowledge of using screen readers, during the debrief, P5 noted "*I was able to effectively use the navigation commands with my screen reader and the system allowed that to happen. Sometimes systems and websites lock them down or don't use good headings so you can't use your navigation commands with your screen reader, which is obnoxious.*" Adopting a different strategy, P2 used the browser's search feature to jump to a portion of the page (e.g., using a label of a bar) and then used nearby headers to orient himself or jump to a different component. While these navigation strategies are not unique to our interface and it is well-known that screen reader users adopt a wide range of navigation patterns [8], these observations highlight the consistent yet flexible nature of the dashboard structure generated by our system.

Change descriptions promote interactive querying. Change descriptions enabled participants to sift through data subsets to identify ones that are of potential interest for deeper investigation. For instance, working with the colleges dashboard (Figure 3A), P1 set his focus to the *State* filter widget and keyed through the different states, listening to the first couple of bullet points about the KPI values. If he heard a state that was of interest to him personally (e.g., New York) or found the KPI values to be interesting, he would navigate down to the change description section using its header to go through the other data facts and then either investigate the dashboard further or return to the filter and move on to a new state. Although we highlight only one example here, this was a pattern we noticed all participants perform, especially in the open-ended exploration task phase. When first introduced to the change description feature in the 'Summary' mode, three participants (P2, P4, P5) also preemptively asked about comparisons, validating our design principle to support multiple modes of analysis. P4, for example, after going through the change description for the state of Texas, said "*So a random thought that just popped in my mind. It will be kind of cool to be able to maybe compare two states.*" Using this as a segue, we introduced the 'Comparison' mode and asked him to update the filter to a state he'd like to compare Texas to. Hearing the updated change description comparing Texas to Nevada, he noted "*Oh, I like the way you worded that. You gave me the increase, and then you tell me we went from here to there. I like that... I really like the way you've done this analysis. You're almost like reading my mind on the the key features that I want to know about all this.*"

The dashboard structure and interaction affords targeted analysis. During the targeted analysis phase that lasted between 26-38 minutes (mean: 35 minutes), three participants (P1, P4, P5) attempted all eight tasks, one participant (P2) attempted seven tasks, and one participant (P3) attempted four tasks. P1 and P5 correctly responded to all questions, P4 had 7/8 correct responses, P2 had 6/7 correct responses, and P3 got 4/4 responses right. For the two incorrect responses, participants misinterpreted the values and noted the wrong category (i.e., responded with a category with lower value instead of higher). Note that although we provide these numbers for context, they are not representative of task performance time and accuracy as they include the time of reading tasks, switching between the browser and text file reading applications like notepad during a task, as well as any discussion during a task. Instead, our goal here is to highlight that with just ∼15min of training, SRUs were able to utilize our dashboards to interactively perform a breadth of analytic tasks that were previously not possible or required working with the raw data in tools like Excel. As they performed these tasks, we also observed participants employ different interaction patterns highlighting the flexibility afforded by the generated interface. For instance, when answering comparative tasks, three participants preferred using the change description in the 'Comparison' mode whereas the other two preferred using the 'Summary' mode and then manually computing the differences. Another example was participants' preferences for intra-chart navigation. We noticed that some participants used filters more proactively to reduce intra-chart navigation whereas other participants preferred navigating across series and marks wihtin charts when investigating specific data categories.

The interface fosters autonomy by helping users generate and answer questions. The onboarding through the initial description coupled with the accessible dashboard structure and change descriptions enabled participants to not only answer given questions but also to ask new questions of their own. Specifically, during the open-ended task phase with the college admissions dashboard (Figure 3A), all participants started the task by going through the description to gain an understanding of the dashboard's contents. Participants would then use the available filters to focus on a data subset that was of personal interest to them (e.g., states their resided in), subsequently comparing that to other subsets of interest (e.g., two participants compared the states of New York and New Jersey). Commenting on how the ability to filter and track updates aided exploration, P2 said "*The controls work great and I was able to play with them and see how I could create new ideas, or new thoughts or new questions.*" Summarizing her overall experience with the dashboard, P5 noted that the presented interface offered access to data at multiple levels of granularity and could make screen reader users more independent, saying "*Just to be able to have access to data at the granularity level that this system gives you access to is great...You have the equivalent experience that someone who could look at that chart would have. You could filter it by different things and know what you're looking at without having to kind of either ask others or just not have the information.*"

Learnability is an inevitable challenge due to a legacy of inaccessible dashboards. All participants except P3 (who had the least familiarity with data tools and visualizations before the study) said they would be able to learn to use the presented dashboard without any external support. However, all participants noted that it does take time to get comfortable with the interface, primarily because the presented features surpass anything they have experienced with dashboards in the past. For instance, noting that it took her some time to grasp the available information and think of potential questions during the exploration phase, P5 said "*exploration isn't natural to me given that I haven't had access to anything accessible before.*" P2 commented on the trade-off between expressivity and learning time, saying "*Really valuable things take time and energy to appreciate. And that's what I look at this as. If this wasn't very valuable, it could have been pretty quick to use it and go on. But this is powerful, so it takes a little time.*" Although the interface was most difficult to grasp for P3, he clarified he found it very valuable to have access to an entire interactive analysis interface but would need more training and practice with visualizations before he could utilize such an interface effectively.

Complementing the subjective feedback listed above, we also asked participants to rate the prototype on a five-point Likert scale. Specifically, the questions focused on aspects such as its support for different analysis scenarios, learnability, and usefulness of the descriptions. Detailed questions and responses are provided in the supplementary material but overall, the participant ratings mirrored the positive subjective feedback with a mode rating of 5/5 and an average of 4.6/5 across six questions (for each question, 1 = most negative rating and 5 = most positive rating).

Besides the aforementioned observations and feedback on specific aspects of the work, one comment by P4 most aptly captured our overarching motivation for designing a dashboard interface for screen readers: "*Compared to having to download data and rerun updates this will save a lot of time but the main reason this is good though, is because I shouldn't have to download the data and do all that. I should be able to have the same access, like everyone else.*" Considering that when talking about his prior experiences during the introduction phase of the study, P4 had mentioned "*I was comfortable with SQL and doing data mining that way, but never visualization. I don't like guys who do much visualization. They keep us kind of on the in the background*," we take the comment about our dashboard interface providing equivalent access to SRUs as a strong validation of the implemented interface design and features.

6 LIMITATIONS AND FUTURE WORK

The collective feedback from our co-design sessions and the subsequent user study helped us validate the general utility and usability of our approach to make visualization dashboards accessible. Generalizing the approach, however, requires a more extensive user study and exploration of other types of dashboard designs beyond grid layouts (e.g., paginated views, infographic-style dashboards with additional image and media components) and widget types (e.g., sliders for numeric filters). Furthermore, while we support basic intra-chart navigation implicitly provided by Highcharts [23], incorporating more expressive navigation and querying techniques (e.g., [48, 57, 64]) at an individual chart-level within the dashboard could foster a richer analytic experience. For instance, leveraging voice-based querying as in VoxLens [48] could allow users to update a chart and subsequently, Azimuth's change description could be used to understand related updates from other linked charts in the dashboard. Besides these near term areas for improvement, our work also highlights higher-level themes for future research at the intersection of accessibility and data visualization.

Investigating mixed-ability collaboration. A point that came up during both the design sessions and the user study was that Azimuth's dashboards reduced dependency on other individuals and promoted self-service discovery and analysis. While we currently focus on the user experience of consuming/interacting with a dashboard, the generated descriptions could potentially also aid mixed-ability collaboration during dashboard authoring. For instance, the change descriptions could summarize data changes based on actions by a collaborator. Furthermore, variants of the layout summary in InfoBox 1 could highlight differences in the dashboard's design (as opposed to only summarizing the current layout). Along the lines of prior work on mixed-ability collaboration for document writing [15, 34] and presentation authoring [38], a rich area for future work is to investigate whether, and how, descriptions facilitate collaborative design and use of visualization dashboards.

Exploring curb-cut effects of dashboard descriptions for onboarding. Although ubiquitously used as data interfaces across domains, dashboards users often feel lost when they first encounter a dashboard [17, 58]. One potential curb-cut effect of the descriptions generated by Azimuth is that they could serve as a dashboard onboarding tool. Specifically, since the descriptions provide information about both the underlying data and the design, they could help users gain a holistic understanding of the dashboard. Examining the utility the generated descriptions for dashboard onboarding while also investigating potential differences in impact based on varying levels of visualization literacy is a compelling direction for future research.

Considering alternative devices and interaction modalities. Given the nascency of research on accessible dashboards, we only considered the context of a desktop/laptop and default keyboard-based interactions. However, both general visualization research (e.g., [31–33]) and work specific to visualization accessibility (e.g., [13, 24, 57, 63]) have shown promise in incorporating alternative devices (e.g., phones, watches, tablets) and modalities (e.g., touch, voice, sound). Even during the user study, two participants (P1, P3) commented that the layout summary in the dashboard description (InfoBox 1) could be particularly useful to get a sense for the content placement in the context of mobile devices. To this end, an open direction for future work is to explore accessible dashboard design and use on alternative devices, experimenting with different input and output modalities.

Developing "auto-insight" systems for dashboards. A number of automated systems have been proposed to extract data facts or insights for singleton visualizations (e.g., [14, 16, 53, 56]). In our work, we reuse strategies systems to generate a 1-to-1 mapping between facts and dashboard components. However, *interactive* dashboards, in particular, allow users to apply multiple filters and identifying deeper insights about data subsets that span beyond those that can be gained from a single chart. Along these lines, one research opportunity for visualization systems is to develop techniques for generating data facts or insights at a holistic dashboard-level. If successful, such descriptions could replace the list of facts in descriptions like InfoBox 1, making the takeaways more high-level as well as contained when the number of components is very large. Given the recent advancements in large language models, a related idea is also to investigate their use for automatically generating descriptions from a dashboard specification and considering the pros and cons of the resulting text (e.g., are the generated descriptions more domain-specific? Is the text too verbose? Do the models hallucinate? If so, when? And how does this impact user perception?).

Understanding and improving dashboard accessibility at scale. Our proposed approach and understanding of accessibility challenges with dashboards is informed by a combination of the co-design sessions and prior research on data and visualization accessibility [26, 46, 52]. However, gaining a formal, quantified understanding of dashboard accessibility challenges calls for a large-scale analysis of dashboards in the wild (e.g., [40]). Furthermore, while systems like Azimuth can guide dashboard design, widespread authoring of accessible dashboards requires incorporating additional tooling for accessibility within visualization systems. One promising research direction, for instance, is the creation of mixed-initiative interfaces that can leverage approaches similar to Azimuth for generating an initial description and dashboard structure but then allow dashboard authors to fine-tune the system suggestions (e.g., the default system generated data facts in descriptions can be reordered or overridden with domain specific insights).

Examining the role of descriptions beyond dashboards. While they include specific components (e.g., charts, KPIs), dashboards can ultimately be considered a subset of dynamic web pages where the contents of the page change based on real-time updates. This similarity raises the question of how we might leverage descriptions similar to those generated by Azimuth in a broader context to orient people to a dynamic web page and subsequently highlight content changes on the page [9]. From an user interface research standpoint, this raises several questions for future work including: How we might best represent a web page's components and their interactions? How can we detect and prioritize changes across components? And how should the list of changes be presented?, among others.

7 CONCLUSION

Visualization dashboards are commonly used to share and analyze data across domains (e.g., business, healthcare, public policy) but are often inaccessible to SRUs. Through an iterative, co-design study with two blind individuals, we identify design goals for creating accessible dashboards and operationalize these goals through a prototype system, Azimuth. Specifically, Azimuth converts a JSON-based dashboard specification into a web-based dashboard that is optimized for screen readers as well as a set of descriptions to aid dashboard comprehension and interaction. We illustrate Azimuth's utility through two example applications and also evaluate the generated dashboards through a qualitative user study. Based on the observations and feedback from the study, we describe how the generated dashboards afford flexible interaction strategies and empower SRUs to not only answer targeted data questions but also explore the data and ask new questions of their own. Finally, reflecting on our design process and the study, we discuss opportunities for future work on exploring curb-cut effects for visualization research, investigating alternative devices and modalities, and designing accessible visualization dashboards at scale.

ACKNOWLEDGMENTS

This work was funded by Tableau, NSF EDA 2009977 and the Center for Research and Education on Accessible Technology and Experiences (CREATE).

REFERENCES

[1] 2023. Bootstrap. https://getbootstrap.com/.

[2] Md Zubair Ibne Alam, Shehnaz Islam, and Enamul Hoque. 2023. SeeChart: Enabling Accessible Visualizations Through Interactive Natural Language Interface For People with Visual Impairments. *arXiv preprint arXiv:2302.07742* (2023).

[3] A Alberts. 2017. Eye-tracking study: 5 key learnings for data designers everywhere.

[4] HK Ault, JW Deloge, RW Lapp, MJ Morgan, and JR Barnett. 2002. Evaluation of long descriptions of statistical graphics for blind and low vision web users. In *International Conference on Computers for Handicapped Persons*. Springer, 517–526.

[5] Benjamin Bach, Euan Freeman, Alfie Abdul-Rahman, Cagatay Turkay, Saiful Khan, Yulei Fan, and Min Chen. 2022. Dashboard Design Patterns. *arXiv preprint arXiv:2205.00757* (2022).

[6] Jenny Joyce Beumer, Ab de Haan, and Josine van der Ven. 2000. Implications of computer-mediated communication for people who are visually impaired in dealing with complex visualization tasks. *Journal of Visual Impairment & Blindness* 94, 7 (2000), 453–456.

[7] Matt Blanco, Jonathan Zong, and Arvind Satyanarayan. 2022. Olli: An Extensible Visualization Library for Screen Reader Accessibility. In *IEEE VIS Posters*. http://vis.csail.mit.edu/pubs/olli

[8] Yevgen Borodin, Jeffrey P Bigham, Glenn Dausch, and IV Ramakrishnan. 2010. More than meets the eye: a survey of screen-reader browsing strategies. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)*. 1–10.

[9] Yevgen Borodin, Jeffrey P Bigham, Rohit Raman, and IV Ramakrishnan. 2008. What's new? Making web page updates accessible. In *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*. 145–152.

[10] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D^3 data-driven documents. *IEEE transactions on visualization and computer graphics* 17, 12 (2011), 2301–2309.

[11] Stephen M Casner. 1991. Task-analytic approach to the automated design of graphic presentations. *ACM Transactions on Graphics (ToG)* 10, 2 (1991), 111–151.

[12] James I Charlton. 1998. Nothing about us without us. In *Nothing About Us Without Us*. University of California Press.

[13] Pramod Chundury, Biswaksen Patnaik, Yasmin Reyazuddin, Christine Tang, Jonathan Lazar, and Niklas Elmqvist. 2022. Towards Understanding Sensory Substitution for Accessible Visualization: An Interview Study. *IEEE Trans. Vis. Comput. Graph.* 28, 1 (2022), 1084–1094. https://doi.org/10.1109/TVCG.2021.3114829

[14] Zhe Cui, Sriram Karthik Badam, M Adil Yalçin, and Niklas Elmqvist. 2019. Datasite: Proactive visual data exploration with computation of insight-based recommendations. *Information Visualization* 18, 2 (2019), 251–267.

[15] Maitraye Das, Thomas Barlow McHugh, Anne Marie Piper, and Darren Gergle. 2022. Co11ab: Augmenting Accessibility in Synchronous Collaborative Writing for People with Vision Impairments. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–18.

[16] Çağatay Demiralp, Peter J Haas, Srinivasan Parthasarathy, and Tejaswini Pedapati. 2017. Foresight: Recommending visual insights. *arXiv preprint arXiv:1707.03877* (2017).

[17] Vaishali Dhanoa, Conny Walchshofer, Andreas Hinterreiter, Holger Stitz, Eduard Groeller, and Marc Streit. 2022. A Process Model for Dashboard Onboarding. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 501–513.

[18] Wayne W Eckerson. 2010. *Performance dashboards: measuring, monitoring, and managing your business*. John Wiley & Sons.

[19] Danyang Fan, Alexa F Siu, Hrishikesh V Rao, Gene SH 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 (TACCESS)* (2022).

[20] Stephen Few. 2006. *Information dashboard design: The effective visual communication of data*. O'Reilly Media, Inc.

[21] Christopher Frauenberger, Julia Makhaeva, and Katta Spiel. 2016. Designing smart objects with autistic children: Four design exposès. In *Proceedings of the 2016 CHI conference on human factors in computing systems*. 130–139.

[22] Bryan Gould, Trisha O'Connell, and Geof Freed. 2008. Effective practices for description of science content within digital talking books. *Guidelines for Describing STEM Images* (2008).

[23] Highcharts.js. 2023. Accessibility Module. https://www.highcharts.com/docs/accessibility/accessibility-module.

[24] Md Naimul Hoque, Md Ehtesham-Ul-Haque, Niklas Elmqvist, and Syed Masum Billah. 2023. Accessible Data Representation with Natural Sound. (2023).

[25] Crescentia Jung, Shubham Mehta, Atharva Kulkarni, Yuhang Zhao, and Yea-Seul Kim. 2021. Communicating Visualizations without Visuals: Investigation of Visualization Alternative Text for People with Visual Impairments. *IEEE Transactions on Visualization and Computer Graphics* 28, 1 (2021), 1095–1105.

[26] NW Kim, SC Joyner, A Riegelhuth, and Y Kim. 2021. Accessible visualization: Design space, opportunities, and challenges. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 173–188.

[27] Younghoon Kim and Jeffrey Heer. 2018. Assessing effects of task and data distribution on the effectiveness of visual encodings. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 157–167.

[28] Richard E Ladner. 2015. Design for user empowerment. *interactions* 22, 2 (2015), 24–29.

[29] Jonathan Lazar, Aaron Allen, Jason Kleinman, and Chris Malarkey. 2007. What frustrates screen reader users on the web: A study of 100 blind users. *International Journal of human-computer interaction* 22, 3 (2007), 247–269.

[30] Bongshin Lee, Eun Kyoung Choe, Petra Isenberg, Kim Marriott, and John Stasko. 2020. Reaching broader audiences with data visualization. *IEEE Computer Graphics and Applications* 40, 2 (2020), 82–90.

[31] Bongshin Lee, Petra Isenberg, Nathalie Henry Riche, and Sheelagh Carpendale. 2012. Beyond mouse and keyboard: Expanding design considerations for information visualization interactions. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2689–2698.

[32] Bongshin Lee, Arjun Srinivasan, Petra Isenberg, John Stasko, et al. 2021. Post-wimp interaction for information visualization. *Foundations and Trends® in Human–Computer Interaction* 14, 1 (2021), 1–95.

[33] Bongshin Lee, Arjun Srinivasan, John Stasko, Melanie Tory, and Vidya Setlur. 2018. Multimodal interaction for data visualization. In *Proceedings of the 2018 International Conference on Advanced Visual Interfaces*. 1–3.

[34] Cheuk Yin Phipson Lee, Zhuohao Zhang, Jaylin Herskovitz, JooYoung Seo, and Anhong Guo. 2022. Collabally: Accessible collaboration awareness in document editing. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–17.

[35] Alan Lundgard and Arvind Satyanarayan. 2021. Accessible Visualization via Natural Language Descriptions: A Four-Level Model of Semantic Content. *IEEE transactions on visualization and computer graphics* 28, 1 (2021), 1073–1083.

[36] Kim Marriott, Bongshin Lee, Matthew Butler, Ed Cutrell, Kirsten Ellis, Cagatay Goncu, Marti Hearst, Kathleen McCoy, and Danielle Albers Szafir. 2021. Inclusive data visualization for people with disabilities: a call to action. *Interactions* 28, 3 (2021), 47–51. https://doi.org/10.1145/3457875

[37] Microsoft. 2022. Microsoft Power BI Data Stories Gallery. https://community.powerbi.com/t5/Data-Stories-Gallery/bd-p/DataStoriesGallery.

[38] Yi-Hao Peng, Jason Wu, Jeffrey Bigham, and Amy Pavel. 2022. Diffscriber: Describing Visual Design Changes to Support Mixed-Ability Collaborative Presentation Authoring. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–13.

[39] Venkatesh Potluri, Tadashi E Grindeland, Jon E Froehlich, and Jennifer Mankoff. 2021. Examining visual semantic understanding in blind and low-vision technology users. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.

[40] Joanna Purich, Arjun Srinivasan, Michael Correll, Leilani Battle, Vidya Setlur, and Anamaria Crisan. 2023. Toward a Scalable Census of Dashboard Designs in the Wild: A Case Study with Tableau Public. *arXiv preprint arXiv:2306.16513* (2023).

[41] Ghulam Jilani Quadri and Paul Rosen. 2021. A survey of perception-based visualization studies by task. *IEEE Transactions on Visualization and Computer Graphics* (2021).

[42] Jonathan C Roberts. 2007. State of the art: Coordinated & multiple views in exploratory visualization. In *Fifth international conference on coordinated and multiple views in exploratory visualization (CMV 2007)*. IEEE, 61–71.

[43] Bahador Saket, Alex Endert, and Çağatay Demiralp. 2018. Task-based effectiveness of basic visualizations. *IEEE transactions on visualization and computer graphics* 25, 7 (2018), 2505–2512.

[44] Alper Sarikaya, Michael Correll, Lyn Bartram, Melanie Tory, and Danyel Fisher. 2018. What do we talk about when we talk about dashboards? *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 682–692.

[45] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2016. Vega-lite: A grammar of interactive graphics. *IEEE transactions on visualization and computer graphics* 23, 1 (2016), 341–350.

[46] Ather Sharif, Sanjana Shivani Chintalapati, Jacob O. Wobbrock, and Katharina Reinecke. 2021. Understanding Screen-Reader Users' Experiences with Online Data Visualizations. In *ASSETS '21: The 23rd International ACM SIGACCESS Conference on Computers and Accessibility, Virtual Event, USA, October 18-22, 2021*, Jonathan Lazar, Jinjuan Heidi Feng, and Faustina Hwang (Eds.). ACM, 14:1–14:16. https://doi.org/10.1145/3441852.3471202

[47] Ather Sharif, Katharina Reinecke, Andrew M Zhang, and Jacob O Wobbrock. 2023. Understanding and Improving Drilled-Down Information Extraction from Online Data Visualizations for Screen-Reader Users. (2023).

[48] Ather Sharif, Olivia H. Wang, Alida T. Muongchan, Katharina Reineck, and Jacob O. Wobbrock. 2022. VoxLens: Making online data visualizations accessible with an interactive JavaScript plug-in. In *CHI '22: CHI Conference on Human Factors in Computing Systems*. ACM. https://doi.org/10.1145/3491102.3517431

[49] Ather Sharif, Andrew Mingwei Zhang, Anna Shih, Jacob O Wobbrock, and Katharina Reinecke. 2022. Understanding and Improving Information Extraction From Online Geospatial Data Visualizations for Screen-Reader Users. In *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility*. 1–5.

[50] Ben Shneiderman. 1996. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE symposium on visual languages*. IEEE, 336–343.

[51] Alexa Siu, Gene SH 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*. 1–19.

[52] Alexa F Siu, Danyang Fan, Gene SH Kim, Hrishikesh V Rao, Xavier Vazquez, Sile O'Modhrain, and Sean Follmer. 2021. COVID-19 highlights the issues facing blind and visually impaired people in accessing data on the web. In *Proceedings of the 18th International Web for All Conference*. 1–15.

[53] Arjun Srinivasan, Steven M Drucker, Alex Endert, and John Stasko. 2018. Augmenting visualizations with interactive data facts to facilitate interpretation and communication. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 672–681.

[54] Tableau. 2023. Build Accessible Dashboards. https://help.tableau.com/current/pro/desktop/en-us/accessibility_dashboards.htm.

[55] Tableau. 2023. Tableau Public. https://public.tableau.com/app/discover.

[56] Bo Tang, Shi Han, Man Lung Yiu, Rui Ding, and Dongmei Zhang. 2017. Extracting top-k insights from multi-dimensional data. In *Proceedings of the 2017 ACM International Conference on Management of Data*. 1509–1524.

[57] John R Thompson, Jesse J 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*. 1–18.

[58] Melanie Tory, Lyn Bartram, Brittany Fiore-Gartland, and Anamaria Crisan. 2021. Finding their data voice: Practices and challenges of dashboard users. *IEEE Computer Graphics and Applications* (2021).

[59] WR Utz. 1979. The Blind Student in the Mathematics Classroom. *The American Mathematical Monthly* 86, 6 (1979), 491–494.

[60] Visa. 2022. Visa Chart Components. https://developer.visa.com/pages/chart-components.

[61] Yun Wang, Zhida Sun, Haidong Zhang, Weiwei Cui, Ke Xu, Xiaojuan Ma, and Dongmei Zhang. 2019. Datashot: Automatic generation of fact sheets from tabular data. *IEEE transactions on visualization and computer graphics* 26, 1 (2019), 895–905.

[62] Steve Wexler, Jeffrey Shaffer, and Andy Cotgreave. 2017. *The big book of dashboards: visualizing your data using real-world business scenarios*. John Wiley & Sons.

[63] Haixia Zhao, Catherine Plaisant, Ben Shneiderman, and Jonathan Lazar. 2008. Data sonification for users with visual impairment: a case study with georeferenced data. *ACM Transactions on Computer-Human Interaction (TOCHI)* 15, 1 (2008), 1–28.

[64] 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 (Proc. EuroVis)* (2022). http://vis.csail.mit.edu/pubs/rich-screen-reader-vis-experiences