

Umwelt: Accessible Structured Editing of Multimodal Data Representations

Jonathan Zong
jzong@mit.edu

Massachusetts Institute of Technology
Cambridge, Massachusetts, USA

Isabella Pedraza Pineros
ipedraza@mit.edu

Massachusetts Institute of Technology
Cambridge, Massachusetts, USA

Mengzhu (Katie) Chen
mzc219@mit.edu

Massachusetts Institute of Technology
Cambridge, Massachusetts, USA

Daniel Hajas
d.hajas@ucl.ac.uk

Global Disability Innovation Hub
Stratford, UK

Arvind Satyanarayan
arvindsatya@mit.edu

Massachusetts Institute of Technology
Cambridge, Massachusetts, USA

Editor

Data Fields Visual Audio

Fields

Select fields: date
 precipitation
 temp_max
 temp_min
 wind
 weather

Key: temp_max
temp_min

precipitation

Type quantitative

Encodings:

▼ Additional options

Aggregate

Bin

temp_max

Type quantitative

Encodings:

▼ Additional options

Bin

temp_min

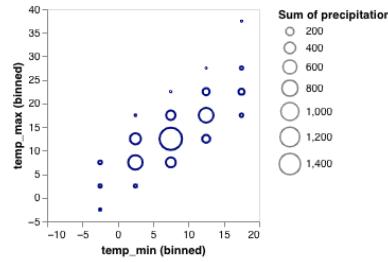
Type quantitative

Encodings:

▼ Additional options

Bin

Viewer



A scatterplot. With axes binned temp_max and binned temp_min. Y-axis titled binned temp_max. For a quantitative scale. With values from -5 to 40. The average value for the binned temp_max field is 14, the maximum is 35, and the minimum is -5. X-axis titled binned temp_min. For a quantitative scale. With values from -10 to 20. The average value for the binned temp_min field is 4, the maximum is 15, and the minimum is -10. Legend titled sum precipitation. For a quantitative scale. With values from 0 to 1600. The average value for the sum precipitation field is 170, the maximum is 1533.20, and the minimum is 0.

binned temp_max
binned temp_min
pitch: sum precipitation
Playback rate x
Playback order

Speak audio axis ticks
Audio axis speech rate

Unmuted

Figure 1: A multimodal data representation designed in the Umwelt interface. Users specify fields and encodings in the accessible structured editor (left), which render into a visualization, textual structure, and sonification in the viewer (right).

ABSTRACT

We present Umwelt, an authoring environment for interactive multimodal data representations. In contrast to prior approaches, which



This work is licensed under a Creative Commons Attribution International 4.0 License.

CHI '24, May 11–16, 2024, Honolulu, HI, USA

© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0330-0/24/05
<https://doi.org/10.1145/3613904.3641996>

center the visual modality, Umwelt treats visualization, sonification, and textual description as coequal representations: they are all derived from a shared abstract data model, such that no modality is prioritized over the others. To simplify specification, Umwelt evaluates a set of heuristics to generate default multimodal representations that express a dataset's functional relationships. To support smoothly moving between representations, Umwelt maintains a shared query predicated that is reified across all modalities — for instance, navigating the textual description also highlights the visualization and filters the sonification. In a study with 5 blind / low-vision expert users, we found that Umwelt's multimodal representations afforded complementary overview and detailed perspectives on a dataset, allowing participants to fluidly shift between task- and representation-oriented ways of thinking.

CCS CONCEPTS

- Human-centered computing → Visualization toolkits; Accessibility systems and tools.

KEYWORDS

multimodal data representation, visualization, textual description, sonification, accessibility

ACM Reference Format:

Jonathan Zong, Isabella Pedraza Pineros, Mengzhu (Katie) Chen, Daniel Hajas, and Arvind Satyanarayan. 2024. Umwelt: Accessible Structured Editing of Multimodal Data Representations. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA*. ACM, New York, NY, USA, 20 pages. <https://doi.org/10.1145/3613904.3641996>

1 INTRODUCTION

For blind and low-vision (BLV) people to be equal participants in interactive data analysis, they must be able to not only consume data representations created by others, but also create their own custom representations by rapidly prototyping and examining alternative designs [16, 20]. Critically, to have full agency over this process, BLV people must be able to independently author and understand data representations without relying on sighted assistance [16]. In pursuit of these goals, accessibility research has begun to investigate multimodal data representation — that is, not only visualization but also textual description, sonification, and other modalities — with initial research results suggesting that the complementary use of multiple modalities can effectively facilitate analysis. For instance, when both sonification and textual description are available, a screen reader user can get a high-level overview from sonification and use it to contextualize their detailed textual exploration [14], and structured textual description helps low-vision magnifier users understand a scatterplot even when data points are visually occluded [40]. Each modality structures information with different spatial and temporal trade-offs, so they often afford different tasks, comparisons, and navigation strategies. Because of these modality differences, a screen reader user might want to author multiple representations to accomplish different goals, and easily switch between representations to develop a more holistic understanding of the data.

Unfortunately, existing tools for creating multimodal data representations center the visual modality: they assume the existence of a visualization that can *then* be converted into an accessible representation. For instance, Chart Reader [37] and VoxLens [33] derive textual and sonified representations from an input specification of a visualization, while the SAS Graphics Accelerator [1] and Highcharts Sonification Studio [9] provide editors for non-visual representations that require users to first specify a visualization. This ordering imposes limitations on both the authoring process and the expressivity of the output representations. In particular, it is challenging for a BLV person to independently create and interpret non-visual data representations unless they can first generate a corresponding visual chart — a problem that pervades existing statistical software [16]. Moreover, a visualization-centric authoring process imposes an undue emphasis on replicating visual affordances non-visually by directly re-mapping encodings, instead of considering the distinct affordances of non-visual modalities. As a result, this approach constrains the set of output representations that systems consider — for instance, Chart Reader and VoxLens restrict their support to a limited subset of chart forms that are straightforwardly amenable to sonification (e.g. bar charts, line charts) while sonifications based on other chart forms (e.g. scatterplots), or that diverge from the original chart's visual encoding (e.g. because they involve data transformations or interactions not specified in the visualization) remain underexplored.

In this paper, we present Umwelt¹: an authoring environment for multimodal data representations designed to de-center the visual modality. A screen reader user can use Umwelt's *structured editor* to specify data representations that include visualization, structured textual description, and sonification. Instead of using a visual specification to generate non-visual representations, Umwelt derives each modality from a shared abstract data model. As a result, users can create these representations in any order and/or specify only a subset of the three modalities as desired. Moreover, via different sections in the editor, a user can switch between editing all modalities simultaneously, or making fine-grained edits to a particular modality. To help users manage the upfront complexity of authoring a multimodal representation, the editor evaluates set of heuristics to generate default representations that express the dataset's functional relationships, and that a user can freely modify. For example, a stocks dataset with a field `price` that depends on independent variables `date` and `symbol` will result in default representations that afford easily looking up the `price` for a given tuple of `date` and `symbol` — a multi-series line chart, a textual structure that can group by `symbol` and `date`, and a sonification that plays back the `price` for each `date`, by `symbol` (Figure 7A).

The editor's state is rendered in Umwelt's *viewer* as independent visual, textual, and sonification views that are interactively linked together. These interactions help maintain a shared context across modalities — for instance, navigating the text structure also

¹The system is named after the concept of *umwelt* in Jakob von Uexküll's semiotic theories [38]. An organism's *umwelt* is the perceptual world produced by its subjective sense experience. Sense faculties vary across species; for example, bats hear ultrasound, and birds sense magnetic fields. As science journalist Ed Yong and disability activist Alice Wong note in an interview [2], the idea of *umwelt* does not support the notion that there is a normative sensory apparatus, either throughout nature or within the human species. Instead, it encourages us to equally value different subjective sense experiences.

highlights the corresponding data visually and filters the sonification domain to only play the selected values – and encourage users to think of the modalities as complementary views into the data. Keyboard shortcuts help a screen reader user quickly move back and forth between the editor and viewer, enabling a tight non-visual feedback loop for confirming the results of edits during prototyping. The editor state is backed by an internal declarative specification (Figure 4). This specification language describes an expressive space of multimodal representations in the viewer. For example, Umwelt extends Vega-Lite’s concept of view composition to express multi-view textual structures and sonifications.

We designed Umwelt through an iterative co-design process involving co-author Hajas, who is a blind researcher with relevant expertise. We evaluate our contribution with multiple evaluation methods, following best practices [28]. Through an example gallery, we demonstrate that Umwelt’s abstractions can express multimodal representations that span a variety of dataset semantics, data types, and view compositions. We also conduct a study involving 5 expert BLV screen reader users to understand how the editor and viewer help users conceptualize, author, and explore multimodal data representations. Our findings surface rich themes about how screen reader users approach multimodal data analysis. We found that participants relied on complementary representations to move between overview and detail, and to manage cognitive and sensory load. Interactive synchronization and runtime customizations enabled participants to access the data by reconfiguring and switching representations to use the one that best suited their immediate needs. Participants also envisioned multimodal representations playing a role in facilitating communication between people who rely on different senses. We also found that the editor reduced challenges associated with specifying representations, and surfaced different ways of thinking about the relationship between specifications and users’ goals.

2 RELATED WORK

Our work is informed by existing approaches to multimodal data representations and systems for accessibly authoring non-visual data representations. In this section, we briefly survey this pertinent literature to better characterize Umwelt’s contributions.

2.1 Multimodal Data Representations

Researchers and practitioners have explored a variety of approaches to data representations beyond visualization. Some systems have focused on one primary alternate modality – for example, Olli [7] explores how textual descriptions can be structured to provide varying levels of detail [40]. A larger body of systems has explored how multiple non-visual modalities can be used in concert. For instance, Apple’s VoiceOver Data Comprehension feature on iOS [11] offers out-of-the-box support for making data accessible through verbal descriptions and sonification (or non-speech audio). Similarly, research systems have explored methods for combining tactile graphics with voice [3, 4], sonification with voice [19], haptics and sonification [15], and sonification and interactive question-answering. Among such multimodal systems, Chart Reader [37] is a particularly apt point of comparison to our work because, like

Umwelt, it incorporates best practices in visualization, structured textual description, and sonification into a single analysis interface.

While these systems make important and necessary contributions to accessible visualization, they share a common assumption: they begin with a visual artifact and attempt to retarget visual affordances to non-visual modalities. For example, Olli, VoxLens, and Chart Reader all require a visualization specification as input to generate their non-visual representations. As a consequence, these systems are often unable to express data representations that do not have an analogous visualization. Chart Reader, for instance, can only express sonifications that directly correspond to the specific typology of chart types it supports. In contrast, Umwelt does not derive its non-visual representations from the visual specification. Instead, its three modalities are treated as equal outputs, all derived from an abstract data model that is shared across modalities. In section 6, we show examples of multimodal representations rendered in the Umwelt viewer that exceed the expressiveness of prior systems (e.g. because their audio specification diverges from the visual).

2.2 Accessible Authoring Tools for Non-Visual Data Representations

In contrast to tools that convert an existing artifact into another representation, researchers have also explored authoring toolkits for multimodal representations. However, most existing toolkits correspond to a single non-visual modality. For example, Highcharts Sonification Studio [9] is an authoring tool for producing charts with sonification, and SVGPlott [13] is an authoring tool for tactile charts. Though these tools are designed to author non-visual data representations, they require a user to specify a visualization to convert into a non-visual form. Consequently, they suffer from the same expressiveness issues discussed in subsection 2.1. For example, SAS Graphics Accelerator [1] includes an authoring workflow that makes charts accessible via sonification and textual description, yet does not support sonification for many chart types.

Because these authoring environments require users to specify visualizations, they impose additional demands on BLV users. For instance, users must have a visual form in mind before creating a non-visual representation, and need an accessible way to verify the accuracy of their visual specification. Umwelt addresses these concerns by allowing users to create representations in any order, and without requiring users to specify all three modalities. Instead, representations are authored independently, reducing the need to conceptualize all outputs in terms of the visual modality. For example, when a user loads a dataset, a textual structure will be generated describing the data in terms of its fields. They can directly specify a sonification by assigning audio encodings, without needing to specify them in terms of visual concepts like the x- or y-axis.

Of existing authoring toolkits, the closest point of comparison is PSST [27], which enables BLV users to create multimodal representations of streaming data that include sonification, spoken description, and physical laser-cut artifacts. Just as we propose with Umwelt, PSST does not require a visual specification. However, PSST differs from Umwelt in terms of its level of abstraction; where Umwelt offers a higher-level workflow and abstractions, PSST exposes low-level abstractions such as event streams, handlers, and

a dataflow graph. This has consequences for how users conceptualize and author representations; in 7.2, we discuss how moving between field- and encoding-oriented specification in Umwelt's editor helped users reason about data in terms of both tasks and representations.

3 MOTIVATION: DE-CENTERING THE VISUAL MODALITY

In this section, we discuss how the overarching motivation of de-centering the visual modality in data analysis translates to concrete design goals for multimodal authoring systems. We then walk through an example usage scenario to demonstrate how instantiating these design goals in Umwelt enables a user to conduct independent data exploration using multiple complementary modalities.

3.1 Design Goals

We designed Umwelt through an iterative co-design process led by co-authors Zong and Hajas. Hajas is a blind researcher with relevant experience in designing accessible representations. Over the course of about a year, we developed multiple prototypes of various interactive sonification and textual description techniques, accessible editor interfaces, and syntax prototypes of Umwelt's abstract model. All co-authors discussed prototypes regularly over Zoom call and email, reflecting on their strengths and weaknesses and brainstorming directions for additional iteration.

Early in the design process, we identified several challenges in the design of an authoring environment for multimodal data representations that arose from the core motivating principle of de-centering the visual representation. We synthesized these challenges into a set of design goals (DGs) that guided our iterative process and influenced team discussions where we reflected on candidate designs. Sections 4 and 5 will elaborate on how these design goals are addressed in the editor and viewer, respectively.

- (1) **Deferred commitment to a modality.** Authors often do not begin a rapid prototyping process with a concrete idea of their desired end state. As such, it is important for an authoring tool to offer the flexibility to easily try many candidate representations in an exploratory manner. In a multimodal system, this might involve freely editing different modalities in any order, or easily changing a field's mappings from one modality to another as they explore possible designs. In existing systems, the non-visual modalities depend on a visual modality. This requires an author to *prematurely commit* [6] to a visual representation before specifying other modalities. Our goal is instead to encourage deferred commitment. For example, an author should be able to specify non-visual modalities independently without first needing to create a visual specification.
- (2) **Complementary use of modalities.** An advantage of a multimodal system is that users are not required to rely on a single representation to meet all of their needs. Due to differences in how each modality conveys information, it is difficult to expect any single representation to act as a standalone replacement for another. Instead, a goal of our system is to encourage users to choose the modality that best

suits their task at any given time, or use multiple modalities together to gain a more complete understanding.

- (3) **Common ground between mixed-ability collaborators.** In a system designed to de-center the visual representation, the visualization still serves an important purpose for collaboration between people who primarily use different senses. For instance, screen reader users are not a monolith; people who use both a screen reader and a magnifier, like many low-vision users do, benefit from referencing a visual chart alongside other representations. Additionally, an important possible use of data analysis is to communicate about it with other people in personal or professional settings. BLV data analysts may need to communicate their findings to audiences that include people with different levels of vision, or participate in discussions where sighted colleagues are referencing visual concepts. Consequently, we find it important that representations establish common ground for diverse collaborators.

3.2 Authoring Multimodal Representations Co-Equally: An Example Usage Scenario

To demonstrate the process of authoring and analyzing multimodal representations in Umwelt, we walk through a scenario in which a screen reader user named Lula explores Hans Rosling's well-known Gapminder dataset [30].

Field-driven default specification. When Lula loads the dataset (Figure 2.1) into the editor in the DATA tab, the FIELDS tab (Figure 2.2) is populated with all of the dataset's fields. Lula inspects the set of checkboxes labeled "select fields" to ascertain that all six fields are initially checked (Figure 2.3) and thus are participating in the default multimodal representation. Reading the section below these checkboxes, Lula learns that Umwelt has inferred a composite key of (year, country) (Figure 2.4). Jumping over to the viewer by pressing the 'v' key on their keyboard, they observe that the initial representation is a textual structure that hierarchically groups and organizes the data for each field (Figure 2.5).

Lula decides that they want to analyze the life expectancy vs. fertility rate of countries over time, mirroring Rosling's original global health scatterplot [30]. They jump back to the editor with the 'e' key and tab through the checkboxes to keep only year, country, life_expect, and fertility checked (Figure 2.6). Umwelt's infers a new multimodal representation by reasoning about the dataset's keys and the measure types of the selected fields (i.e., nominal, quantitative, etc.). For the fields Lula has selected, as the data is keyed by (year, country), Umwelt assumes that a typical reader will use these fields to lookup the value fields (life_expect, fertility).

Though there may be multiple ways to represent the same key-value semantics, the goal of Umwelt's heuristics is to provide an initial starting point rather than a single best representation. Thus, for the selected fields, Umwelt produces a multimodal representation that includes a small multiple of connected scatterplots (Figure 2.7), a textual structure (Figure 2.8), and two sets of audio controls (Figure 2.9), such that all modalities support this lookup operation via their modality-specific affordances. The visualization facilitates the lookup by faceting the data into multiple views by the country field, and using the year to order a connected scatterplot in each

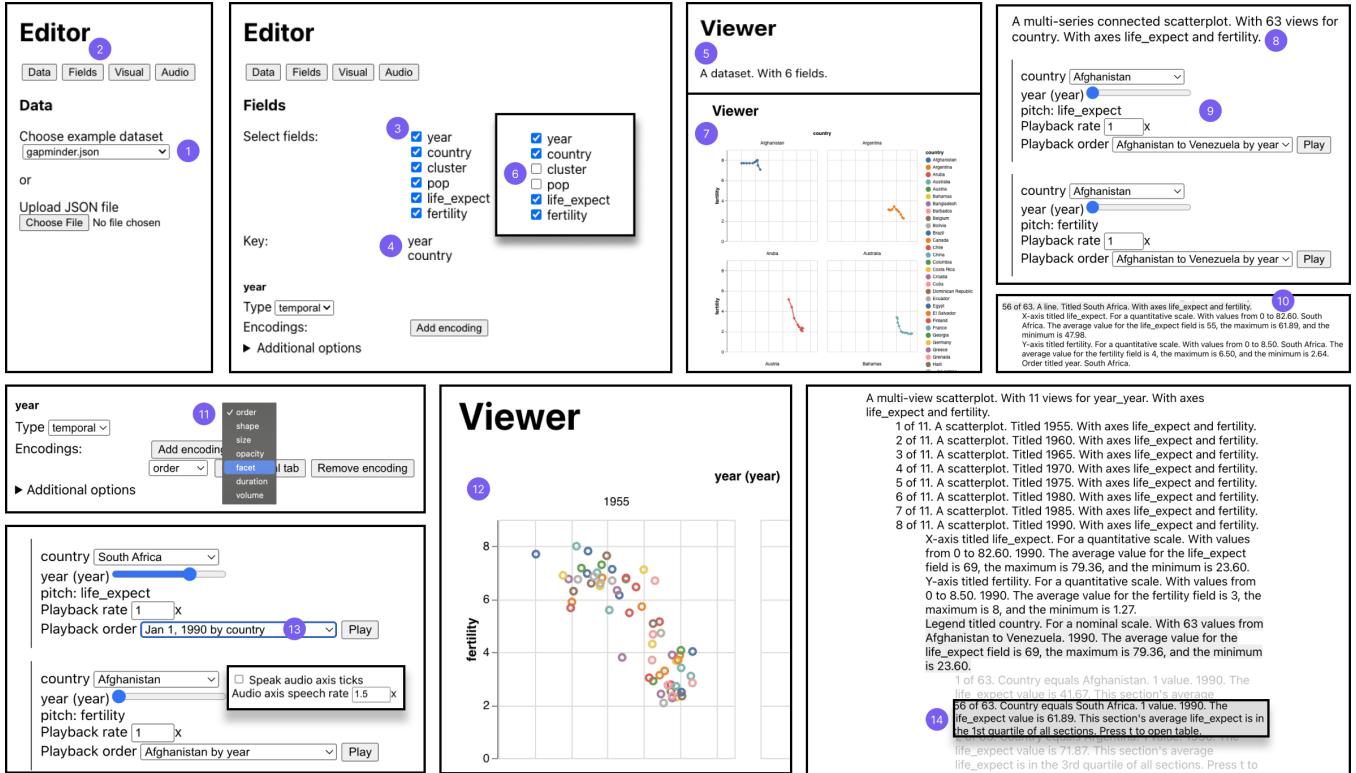


Figure 2: An analyst's workflow in Umwelt: 1–6 illustrate an analyst's process of creating an initial multimodal data representation (shown in 7–9). 10 shows their initial exploration, before (11) making edits and then (12–14) continuing their analysis .

view. The sonification supports this same lookup by offering two sets of audio controls – each corresponding to an *audio unit*, or a single audio track that plays a continuous tone with pitch corresponding to `life_expect` or `fertility` respectively. Both audio units allow Lula to select (`year`, `country`) tuples, either via sequential playback by pressing the play button, or via interactive selection by manipulating a dropdown and slider. Finally, the textual structure facilitates the key-value lookup via its hierarchical structure. The first level below the root allows Lula to choose a `country`, and the next level allows Lula to drill further down into `year`, `life_expect`, or `fertility` (Figure 2.10). This hierarchy is generated based on the fields and key, and would exist even without a visual representation; however, because visual information is present, it's used to annotate the fields' descriptions in terms of x, y, and order encodings. At the lowest levels of the tree, Lula receives summary information about the selected data, and they can also press 't' to open a tabular view of the data.

Analyzing data using multiple modalities. Lula starts by pressing 'p' on their keyboard to play the sonification for `life_expect`. Umwelt orients them via *audio axis ticks*, which are spoken announcements of data values interleaved with the sonification to communicate playback progress. The system announces the first country, Afghanistan, and as the sonification progresses, the system speaks the year value prior to the sonified tone for each 5 year interval. After listening through a few country values, Lula understands how to interpret the sounds, and disables the audio axis

ticks feature to more rapidly get a gist of the rest of the data. They observe that `life_expect` generally increases for most countries, but the min and max values can vary widely. For a few countries that sound different from the rest, Lula pauses the sonification by pressing 'p' on their keyboard again. To determine which country they were listening to, Lula tabs to the set of audio controls for `life_expect`, and inspects the `country` dropdown menu which reflects the current position in the paused playback.

Noticing that South Africa's `life_expect` sonification peaks in the middle before dropping again, Lula jumps to the textual structure with the 'o' key and navigates to the corresponding node by using the down arrow to move from the root level to the `country` level, and using the left and right arrow keys to find the node representing South Africa. Descending a level to the x-axis, they read the exact average, min, and max summary values of `life_expect` for South Africa – grounding the sonification they heard before in concrete numbers. By navigating to a sibling branch of the textual hierarchy, Lula is instead able to step through each year to read the exactly value of `life_expect` from 1955 to 2005. To remind themselves of the overall trend, they press 'p'. Their position in the textual structure emits a query predicate that filters the sonification domain and highlights the corresponding data in the visualization. Because their cursor focus remains on the textual description for South Africa over a particular set of years, the sonification only plays through this data subset. Thus, they are able to identify that the peak they heard was for 1990, when `life_expect` was 61.89.

Editing the representation design. Umwelt's heuristics for these four fields prioritize using country as the “outermost” key, or the key that is used first in the composite key lookup operation. In other words, the multimodal representations afford looking up a specific country before exploring it by year. Now that Lula has done that, they may want to explore the data other way: by picking a year and exploring it by country.

To do so, Lula jumps back to the FIELDS tab of the editor, and removes the facet encoding from the definition for country. Umwelt updates the visualization to a single view containing a multi-line connected scatterplot (also known as a trace visualization [29]). The textual descriptions update to reflect this change, though the hierarchical structure remains unchanged; the sonification, similarly, does not change. Lula then updates the definition for year, switching the order encoding for a facet encoding instead (Figure 2.11). As a result, the visualization is faceted by year and each facet contains a scatterplot (Figure 2.12) with one point per country. The textual hierarchy now updates with year at the first level, and country nested underneath. The sonification still does not change as its traversal ordering is based on the key, which has remained the same throughout. Lula is able to verify these edits had the intended effect by quickly jumping back and forth between the viewer and editor with the ‘e’ and ‘v’ hotkeys.

Lula then repeats their preferred analysis process of sonification overview followed by detailed textual exploration. To listen to the life_expect values for each country in a given year, they first select a year using the slider – they start with 1990 to see what else was happening in the world during their previously observed notable year. Then, using the “playback order” dropdown, they select “1990 by country” (Figure 2.13). After turning “speak audio axis ticks” back on and pressing play, they hear the name of each country followed by a short tone corresponding to its life_expect in 1990. This gives them a general sense of the variability of life_expect values in 1990. Listening for South Africa, they have a sense of the relative position of that tone compared to the higher or lower tones that they've heard. Returning to the textual description, they navigate to 1990 and drill down into the country legend. They navigate to South Africa and are reminded of its average life_expect, and read that this value is in the 1st quartile of life_expect values – meaning that it is below the 25th percentile of values (Figure 2.14).

Summary. Using Umwelt, Lula was able to author multimodal data representations involving visualization, structured textual description, and sonification as part of a self-guided exploratory data analysis of the Gapminder data. Using heuristics that account for fields’ measure type and the dataset’s keys, Umwelt generated smart default specifications to help the analyst quickly get started without needing to think deeply about low-level specification across three modalities. Using an overview and detail strategy, Lula started by listening to the sonification, and contextualized what they heard with concrete data values by moving to the corresponding location in the textual hierarchy. This process of smoothly moving between modalities allowed them to leverage the distinct affordances of each modality in a complementary way. As they progressed, they recognized that their emergent goals during analysis would benefit from a change in the representations’ affordances. By making a small number of atomic changes in the editor, Lula was able to generate

a new textual hierarchy and adjust the sonification playback order to explore the data a different way.

4 THE UMWELT EDITOR

With Umwelt’s editor, users specify multimodal representations using an interface designed primarily for screen readers. The editor’s internal state consists of a declarative JSON structure as shown in Figure 4. In this section, we first introduce key parts of the editor, including its main components and its default specification heuristics. Then, we discuss our design rationale and how it addresses our design goals.

4.1 Components of the Umwelt Editor

Umwelt’s editor, as shown in Figure 3A and 3B, is split across four tabs. This organization is motivated by screen reader affordances, and navigation and wayfinding principles. In our co-design process, we found that when a screen reader user wanted to move back and forth between the editor and viewer with their screen reader, it was more difficult to maintain their position if the editor had too much extraneous content visible at once. Organizing the interface into tabs helps screen reader users think about what functionality they need at a given moment, and helps manage page length and the depth of the information hierarchy.

DATA Tab (Figure 3.A.1). A user begins by either loading a tabular dataset or choosing from a pre-populated list of example datasets. Umwelt then performs some simple type inference, and populates the other tabs with the dataset’s fields.

FIELDS Tab (Figure 3.B). This tab lists all the fields in the dataset, with corresponding checkboxes to allow a user to pick which fields should participate in the multimodal representation. When a user checks or unchecks a field, the system evaluates a set of heuristics (described in 4.2) to produce a default multimodal representation. For each selected field, the editor provides a set of controls (Figure 3.B.3) to edit the field’s inferred measure type, groupings, and transforms that may be calculated on the field (e.g. aggregation, binning). These definitions serve as a shared default across all modalities – defaults that can be overridden in modality-specific ways under the appropriate tab (described below). This tab also collates together the encodings a field is participating for both visual and audio modalities, offering user’s a cross-modality perspective that can be important for generating cohesive and complementary experiences as we describe in § 4.3.1.

VISUAL Tab (Figure 3.A.2). This tab allows a user to make edits that apply only to the visual modality. A visual specification includes the visual-specific concept of a mark, and the encodings for that modality that were assigned in the FIELDS tab. Changes to a field definition (e.g. its transforms) apply only to the corresponding visual encoding. To allow users to be able to express multi-view displays (e.g., layered views or small multiples), Umwelt groups a mark and set of visual encodings into a *visual unit* – a concept Umwelt inherits from Vega-Lite [31]. Users can create multiple visual units, which can then be composed together as layers (where units are plotted one on top of the other) or as a concatenation (where units are laid out side-by-side).

AUDIO Tab (Figure 3.A.3). This tab allows a user to make edits that apply only to the audio modality. An audio specification

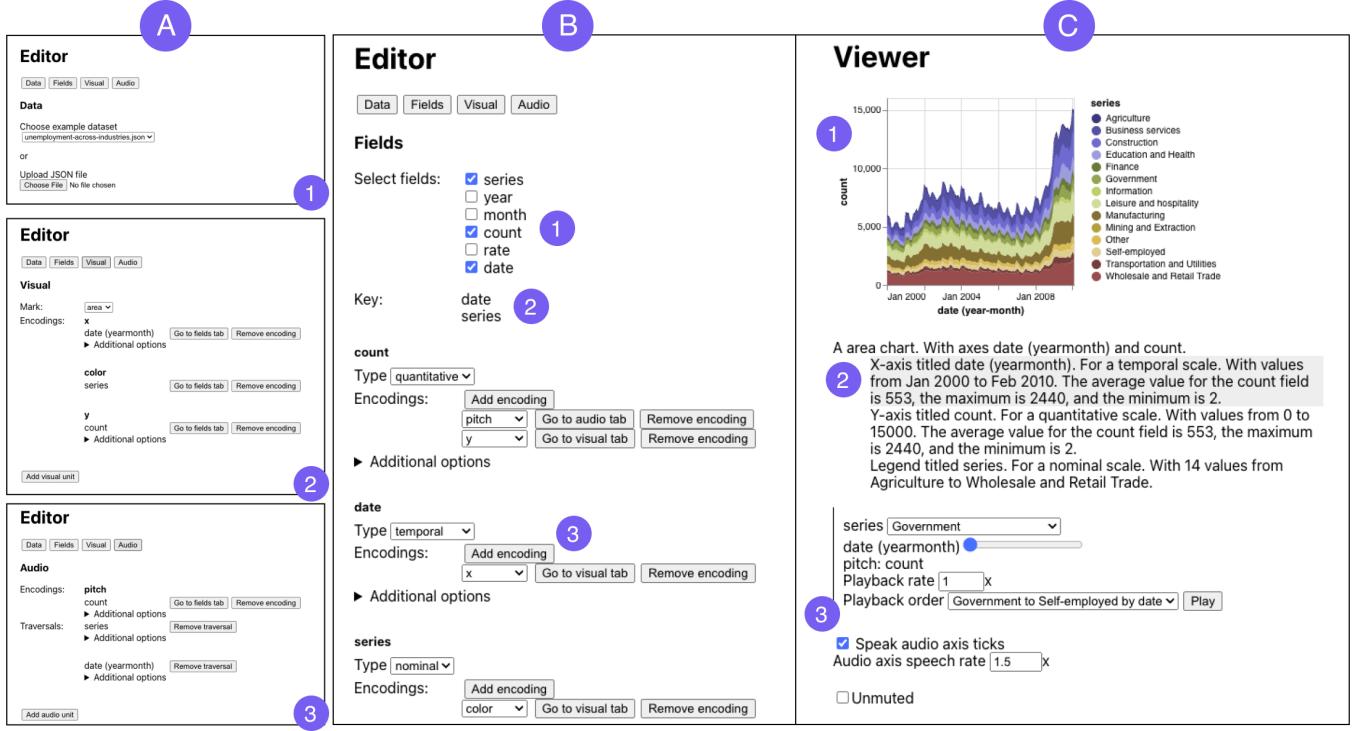


Figure 3: The Umwelt interface. A) The data, visual, and audio tabs of the editor. B) The editor's fields tab, where users specify field definitions and encodings. C) The viewer, where users analyze data with interactive multimodal data representations.

includes encodings from the FIELDS tab, which can be overridden, and *traversals*, an audio-specific abstraction we introduce to control the order in which data points are sonified.

While some visualization systems such as Vega-Lite offer an *order* visual encoding channel (and Tableau offers similar functionality via its *detail shelf*), this channel need only be used in special circumstances—for instance, to control the order that line segments are drawn as part of a connected scatterplot, or to determine z-axis and stack ordering. In contrast, ordering is much more central to the audio modality as data must be linearized into a fixed playback order, and different orderings afford different data lookups and comparisons. For instance, a reader of the stacked area chart in Figure 3.C may want to compare all values for year, one series at a time, to understand the trend of count within each series; or, they may want to traverse all series for a single date before moving onto the next date, to compare which series had the largest count at each date. A visualization reader could easily do both of these operations on the same chart. However, unlike the visual modality, a sonification can only afford one of these operations at a time—switching between the two requires re-ordering the data. Therefore an explicit specification of traversal is required.

A traversal specification is an ordered list of field definitions. The ordered list represents the precedence of groupby operations over the data, which are used to determine a linearized playback order. Consider the Figure 3 example again. The editor state in Figure 3A.3 defines a traversal [series, date]. This means that the data is first grouped by series before date. In the corresponding viewer state

in Figure 3C.3, when a user presses play, the sonification will select the first value of series and iterate through all values for date. At each step, the current tuple of (series, date) values is used to query the value of count to encode it as pitch. After traversing all date values for the given series, the playback advances to the next value for series and iterates over all date values again. Consider the alternate traversal definition of [date, series], which reverses the order in which the fields are grouped. In this case, the sonification would instead start with the first value for date and iterate through all values for series before proceeding to the next date. These two possible traversal specifications correspond to the two use cases described in the previous paragraph.

As this example demonstrates, it sometimes takes multiple sonification specifications to reproduce functionality afforded by a single visualization. To make it easier to provide multiple alternate sonifications, we also extend the concept of view composition to sonification. Like a unit visualization, a *unit sonification* contains a single set of encodings and traversals. Each audio unit corresponds to a single audio track that maps data to a tone, varying its properties (e.g. pitch, volume) according to the specified encodings. Concatenating two unit sonifications means providing two separate, independent audio playback controls side-by-side. A user can move between them to control which one they are listening to, and only one audio unit can be playing at once. Layering two unit sonifications means that they share a traversal and that their encodings are expressed through two audio tones playing simultaneously.

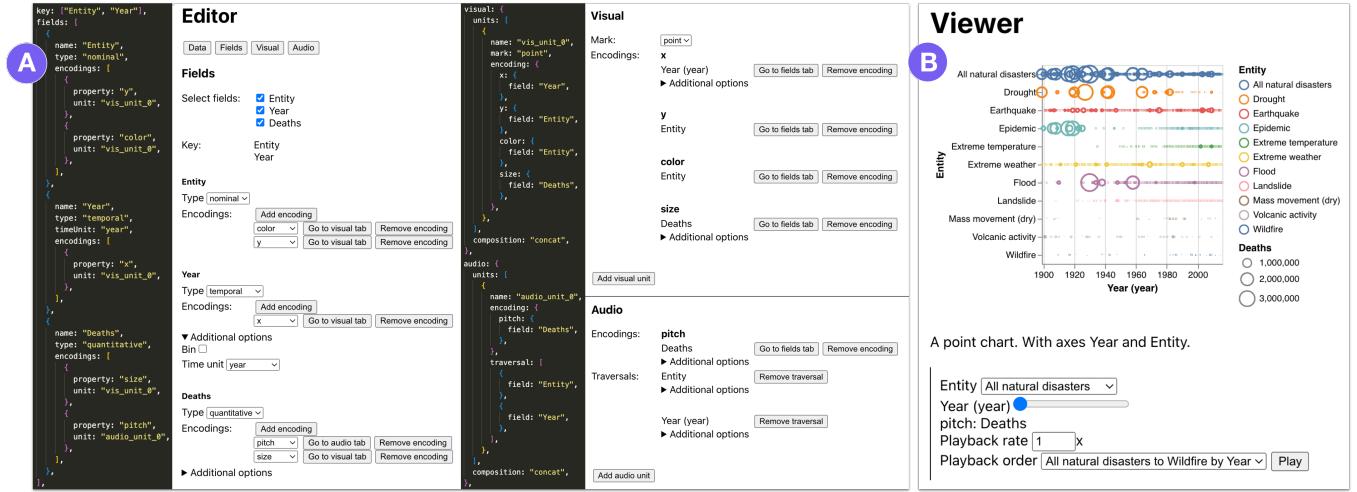


Figure 4: A) Fragments of an internal declarative specification shown next to their corresponding Umwelt editor states. B) The output multimodal representation for that specification.

Future Work: Extending to Text and Other Modalities. Each modality is specified independently, yet each specification inherits from Umwelt’s shared field definition. As such, we expect it will be relatively straightforward to extend Umwelt’s editor to support additional modalities (e.g., textual descriptions, tactile graphics, haptic feedback, etc.) provided there are well-defined abstractions and specification languages for these modalities. For now, although Umwelt currently supports textual output, we have chosen to not offer a TEXT tab as this remains a nascent research area without settled consensus on suitable abstractions. Instead, we have opted to preserve consistency between the textual structure and the visualization. We explain our rationale for this choice, and how it indicates a need for future research, in 4.3.2.

4.2 Default Specifications and Heuristics

Umwelt uses a set of simple heuristics to generate default multimodal representations based on a dataset’s typings and key. In doing so, Umwelt seeks to avoid presenting a user with a blank slate whenever possible, and to accelerate a user in producing commonly used multimodal representations. Once the heuristics are evaluated, a user can modify the resultant defaults via the editor interface. These non-exhaustive heuristics are simple if-else statements that map combinations of field types and primary keys to specification fragments, which we document in Table 1. In our example gallery (Figure 7), examples A, B, C, G, and F were generated by Umwelt’s heuristics while the rest required manual specification.

Our heuristics are motivated by the idea of *functional dependence* between fields in a dataset. In database theory, a functional dependence is the relationship described by a dataset’s primary key — a set of fields whose values uniquely index all rows of the dataset [10, 26]. Just as search algorithms over relational databases use the key to perform data lookups [10], an analyst using a data representation will often implicitly use the key to look up functionally dependent fields (also known as value fields). For example, a common way to read a single-series line chart is to choose a value for

the x-axis field to look up a value for the y-axis field. Though some existing visualization systems, including Tableau, use key-value relationships to model visualizations [36], this concept is even more central to Umwelt because it provides a shared basis for expressing a dataset’s semantics across multiple modalities.

We identified and validated our heuristics through our co-design process, manually authoring specifications for a diverse range of datasets with differently arrangements of typings and key. We worked to identify commonalities between our designs and try to articulate our intuition for why certain representations made more sense than others. For instance, we felt that the stacked area chart in Figure 3.C would be nonsensical if the count were encoded as color, and series as y. This can be explained by the functional dependence of count on date and series. In the sonification case, the key constrains which fields should be encoded at all; we were almost always interested in mapping a value field to an encoding property like pitch and using the key to determine the order of playback. For example, it does not make sense to sonify date or series in Figure 3.C — again, because count is functionally dependent date and series. Finally, in the case of text, the key imposes constraints on the hierarchical structure. In Figure 3.C, we were more likely to want to group by date and series to look up a count value than, for example, to look up a date by first finding its corresponding count.

Table 1: Default specification heuristics based on a dataset's key and typings. T = temporal field, N = nominal field, Q = quantitative field. Each row represents a rule that matches a dataset's key and value tuples. The visualization, sonification, and textual description columns show default specifications for each rule, represented in YAML format for conciseness.

Key	Value	Visualization	Sonification	Textual Structure	Example
T, N	Q	mark: "line" encoding: x: field: t_key y: field: value[0] color: field: n_key	encoding: pitch: field: value[0] traversal: - field: n_key - field: t_key	groupby: n_key children: - groupby: t_key - groupby: value[0]	Figure 7A
T, N (>5 categories)	Q	mark: "point" encoding: x: field: t_key y: field: n_key color: field: n_key size: field: value[0]	encoding: pitch: field: value[0] traversal: - field: n_key - field: t_key	- groupby: t_key - groupby: n_key - groupby: value[0]	Figure 7G
-	Q, Q, N	mark: "point" encoding: x: field: q_value[0] y: field: q_value[1] color: field: n_value	- encoding: pitch: field: value[0] aggregate: "mean" traversal: - field: value[1] bin: true - encoding: pitch: field: value[1] aggregate: "mean" traversal: - field: value[0] bin: true	- groupby: q_value[0] - groupby: q_value[1] - groupby: n_value	Figure 7B
T	Q, Q	mark: "line" encoding: x: field: value[0] y: field: value[1] order: field: key[0]	- encoding: pitch: field: value[0] traversal: - field: key[0] - encoding: pitch: field: value[1] traversal: - field: key[0]	- groupby: key[0] - groupby: value[0] - groupby: value[1]	Figure 7C
T, N, N	Q	mark: "point" encoding: x: field: value[0] y: field: n_key[1] color: field: t_key facet: field: n_key[0]	encoding: pitch: field: value[0] traversal: - field: n_key[0] - field: n_key[1] - field: t_key	groupby: n_key[0] children: - groupby: value[0] - groupby: n_key[1] - groupby: t_key	Figure 7F
T, N	Q, Q	mark: "line" encoding: x: field: value[0] y: field: value[0] facet: field: n_key color: field: n_key order: field: t_key	- encoding: pitch: field: value[0] traversal: - field: n_key - field: t_key - encoding: pitch: field: value[1] traversal: - field: n_key - field: t_key	groupby: n_key children: - groupby: value[0] - groupby: value[1] - groupby: t_key	Figure 2.7

4.3 Design Rationale

4.3.1 Field- vs Encoding-Oriented Specification. The design of a specification language can impose constraints on a user's order of operations [6]. Conventional visualization grammars, including Vega-Lite, are *encoding-oriented*: encoding is a top-level abstraction in a Vega-Lite unit specification, and field definitions are nested within encodings. However, a consequence of encoding-oriented specification is that users must first decide what encodings they are using before assigning fields to them, requiring them to have visual idioms in mind when initially formulating their goals. This limitation is even more pronounced in the context of multimodal representations, as an author may not even have an initial choice of modality in mind. When we began designing Umwelt, we first designed it as an encoding-oriented declarative JSON language. However, our co-design process led us to explore *field-oriented* specification as an alternative: fields are top-level entities and encoding definitions are nested within fields. We felt that a field-oriented approach was amenable to multimodal authoring because an author can make localized changes to a single field definition used across multiple modalities, or switch a field's encoding from one modality to another. The increased ease of these changes enables deferred commitment to a specific representation (DG1). Comparing encoding-oriented and field-oriented specification using the cognitive dimensions framework [6], we argue that field-oriented specification reduces *viscosity* (difficulty of making changes) and increases *provisionality* (ease of exploratory prototyping).

Although field-oriented specification helped us address one of our design goals, we found that it became much more difficult to understand a specification without using modality-specific abstractions. Encoding-oriented specification is prevalent amongst existing declarative grammars because its syntax captures an important semantic property of the relationship between encodings and fields — namely, that each encoding property can only have one field assigned to it. This in turn enables the concise expression of other top-level abstractions: for instance, a unit visualization has one mark and one set of encodings. When reading a Vega-Lite spec, it is easy to understand that a mark and a set of encodings are associated together because they are contained within the same unit spec. When we switched to a field-oriented language, we found that modality-specific definitions became fragmented across field definitions. Consider the example in Figure 5. The encoding-oriented specification in 5A uses unit specs to group the functionality of each modality together. But in 5B's field-oriented specification, encodings belonging to the same unit specification are nested under multiple field definitions. Further, in 5B, modality-specific concepts like mark or traversal are not nested under any individual field, so additional verbosity or repetition must be introduced to associate these concepts with their respective units. In terms of cognitive dimensions, field-oriented specification introduces *role-inexpressiveness* [6] because it is more difficult to read a specification and clearly understand relationships and dependencies between entities affecting the same modality.

Field-oriented and encoding-oriented approaches both had affordances that felt essential but were in conflict with each other in a textual language, leading to significant tension in our design process. Our co-design process led us to bridge between field- and

encoding-oriented specification by designing Umwelt primarily as a structured editor interface, rather than as a textual JSON language. In the editor, the FIELDS tab allows a user to create a field-oriented specification by populating a field with encodings from any modality. Then, the user can switch to the VISUAL or AUDIO tabs to edit modality-specific attributes like mark and traversal, or perform actions that are scoped to one modality (like adding or removing a unit spec). In its internal representation (shown in Figure 4), Umwelt maintains both field-oriented and encoding-oriented abstractions. It links the two kinds of specification together via references. In our prototype language designs, expressing these references in a textual specification language was unwieldy and lead to repetition, but they are suitable for an interface where a user can easily navigate between two views into the same underlying spec. Our eventual design for Umwelt prioritizes field-oriented specification to encourage ease of switching between modalities during exploratory specification, but also allows users to switch to encoding-oriented specification for detailed control.

Designing Umwelt as a structured editor also introduces additional benefits. An editor interface can reactively update the options it presents to a user based on its current state and can, thus, hide operations that would lead a user to invalid states. As a result, each atomic edit in the editor is a transition from one valid specification to another. In contrast, with a textual language, any time a user is partway through typing out a statement, the program will not compile. In terms of the cognitive dimensions of notation framework [6], we would say that an editor interface reduces *error-proneness* compared to the textual language, and affords users a better ability to *progressively evaluate* [6] the specification they are editing.

4.3.2 Aligned vs Disjoint Modalities. In a multimodal data representation, how each representation relates to the others can reflect different design priorities. For example, modalities can be *aligned* in that they redundantly encode the same information, emphasizing a cohesive insight or set of possible comparisons. Or, modalities can be *disjoint*, conveying different aspects of the data that can be synthesized together into a greater whole than the message of each individual representation. In existing systems that derive non-visual representations from the visual, the derived representations are inherently aligned with the original. But in systems like Umwelt where modalities are independent, it can be up to the author's discretion whether modalities are aligned or disjoint.

In our co-design process, thinking about aligned and disjoint modalities uncovered a tension in our design goals, where we seemingly could not simultaneously prioritize DG2 and DG3. On one hand, using visualization and sonification as *disjoint* modalities meant that we could use sonification to focus on comparisons between fields that are difficult to compare visually, or encode fields that are not present in the chart because it would be too visually overwhelming to include them. This additional expressiveness contributes to DG2, where a user can gain additional information from the use of multiple modalities together. On the other hand, using visualization and text as *aligned* modalities preserves consistency between the two representations, which is crucial for BLV users who need the textual representation to access the visualization. During the authoring process, a screen reader user needs the representations to align to verify that they are creating sensible visuals.



Figure 5: Prototypes of encoding- and field-oriented specifications of a scatterplot with concatenated sonification, illustrating the *role-inexpressiveness* [6] of field-oriented textual specification. Color-coded spans on the left side of each text prototype show the lines of code that pertain to each modality: green represents visual while blue represents audio. A) Encoding-oriented specification groups each modality into unit specifications. B) Field-oriented specification fragments each unit's encodings across the spec.

This verification process is crucial to DG3, establishing common ground between blind and sighted users.

In sum, the textual structure can serve a dual purpose of (1) textually conveying the data and (2) making the visualization accessible. These two purposes fulfill DG2 and DG3, respectively, but it is difficult to fulfill both purposes simultaneously because one implies a disjoint representation while the other implies an aligned representation. While Umwelt could allow authors to override field definitions in the textual modality, this would cause the visualization and textual description to become disjoint. Our co-design process led us to prioritize aligned visual and textual representations, and we made a decision not to expose a TEXT tab in the editor. Nonetheless, disjoint visual and textual representations is an important area for future design exploration. For instance, researchers could explore ways to enable a user to customize whether a textual structure is aligned or disjoint on-the-fly.

5 THE UMWELT VIEWER

Umwelt's viewer renders interactive multi-modal representations specified in the editor, including a visualization, a structured textual description, and a sonification. In this section, we first introduce the viewer's components and its linked interaction model. Then, we discuss our design rationale and how the viewer addresses our design goals.

5.1 Multi-Modal Data Representations

The Umwelt Viewer, as shown in Figure 3C, consists of three components: a visualization, a textual structure, and a sonification. Though there is no explicit interaction specification in the editor, each representation in the viewer is implicitly interactive. This interaction-first approach to the design of the viewer is motivated by the need to selectively attend to data. Interactive representations

enable a user to select a subset of data and share that selection across multiple representations. Here, we describe each representation before discussing their interactive behavior in more detail in 5.2.

Visualization. Umwelt converts its internal representation into a Vega-Lite [31] specification to render a visualization (Figure 3.C.1). It augments this specification with additional Vega-Lite selection parameters, resulting in a visual representation that is interactive by default. For example, a user can drag on the visualization to select a rectangular region of data.

Textual structure. Umwelt renders a structured textual description (Figure 3.C.2) with Olli [7], an open-source library that implements Zong, Lee, Lundgard et al.'s design dimensions for screen reader experiences [40]. Olli outputs a hierarchical structure in the shape of a tree. Each node in the structure is associated with a textual description. The root of the structure gives an overall description of the data, while deeper levels in the structure apply successive filters on the data to give more granular descriptions.

The textual output does not require a visual specification, but can use visual information to augment its structure and descriptions. When there is a visual specification present, Olli structures the tree based on the visualization's encodings, and reference visual concepts in its description. As we discussed in 4.3.2, this makes the visual representation accessible for screen reader users and establishes common ground. On the other hand, when there is no visual specification, Olli outputs a relatively flat structure that allows a user to group the data by each field, and uses descriptions that do not reference visual concepts.

Sonification. Umwelt implements an interactive sonification runtime to render its audio specifications (Figure 3.C.3). For each audio unit specification, Umwelt renders a set of audio controls representing a single audio track. A user can press the play button

(or the ‘p’ key on their keyboard) to play and pause the sonification. They can also interactively control their position within the sonification using input elements (i.e., dropdown menus for nominal and ordinal fields, and sliders for quantitative and temporal fields).

To help users keep track of their position in the sonification playback, the sonification runtime uses spoken announcements of data values interleaved with the sonification to communicate playback progress (exposed in the interface as an option called “audio axis ticks”). For example, in Figure 3C, as the sonification traverses date values, the system will speak the date value (e.g. Jan 2000) before playing the sonified segment between each axis tick. If a visualization is present, these ticks will always correspond to the visual ticks for consistency. Otherwise, they are determined by binning the key fields’ domains to calculate regularly spaced intervals (or, for categorical fields, directly reading the value corresponding to each category).

Playback order is an important consideration for sonification, because different playback orders can facilitate different comparisons. For instance, in Figure 3C, playing through all date values for a given series before advancing to the next series is akin to the visual operation of reading each line left to right, getting a sense of each line’s trend. On the other hand, playing through all series for a given date before advancing to the next date is akin to vertically comparing the values for a given x-axis position. Because the preferred order will depend on a user’s goals, Umwelt determines the initial playback order by the ordering of the traversal specification and provides a dropdown menu to select an alternate playback order.

5.2 Coordinating Modalities with Linked Interactions

Though each modality maintains its own interactive state, Umwelt links interactions across modalities to aid analysis. Each modality has one or more interactions that define a selection over the data, and can be modeled as query predicates. For instance, a user can drag a brush over the visualization, navigate to a location or define a custom filter in the textual structure, and navigate to a position in the sonification playback. When a user performs one of these interactions on a representation, that representation updates its own state and emits a query predicate to the other representations. Each representation then reifies this predicate as some sort of effect (e.g. a transformation). Figure 6 shows an example of this process, driven by the textual modality. Olli associates a query predicate with each node in its structure — as a user navigates through the structure, the current node’s predicate describes the data selected by the user’s current position. In this example, a user navigates to a node corresponding to the predicate `{field: 'symbol', equal: 'AAPL'}`. This interaction emits the predicate to the visualization and the sonification. The visualization updates to visually highlight the selected data, and the sonification filters its domain to match the selected data.

5.3 Design Rationale

5.3.1 Highlighting vs Zooming in Non-Visual Representations. In visualization, the same user interaction could plausibly map to multiple possible effects. For example, dragging a rectangular area

on a Vega-Lite scatterplot could *highlight* the data by giving it a conditional encoding (e.g., showing highlighted points in a different color). Or, that same drag interaction could *zoom* into that data (e.g. in an overview + detail interaction), resizing the viewport to only contain the selected data.

Analogously in non-visual modalities, there are multiple possible techniques for conveying the result of an interaction. Consider an example in the textual modality, which we surfaced while prototyping ways to filter a textual structure. One way of applying a filter to a textual structure is to re-scale the structure to fit the filtered data. For instance, an x-axis that originally represented a domain of 0–100 by splitting it into five nodes representing increments of 20 might be re-scaled to split a filtered domain of 50–70 into four nodes representing increments of 5. On testing this approach, co-designer Hajas compared this feature to “zooming in” on a visualization by changing its viewport. Another way of applying a filter is to leave the structure unchanged while re-flowing the structure with only the filtered data. For instance, the previous example would still have five nodes representing increments of 20, but many of the nodes would be empty after applying the filter. This approach is more analogous to “highlighting” a visualization, because the viewport remains the same but the un-selected data is de-emphasized.

Though zooming and highlighting appear to be recurring concepts across modalities, it is not clear that either is universally preferable. Currently, Umwelt’s visual representation uses highlighting to convey interactive state because this is a more common interaction technique in visualization. This makes sense when considering the fact that visually, it is helpful to maintain a consistent viewport to situate a highlighted subset within the broader context of the full data. However, in our co-design process, we felt that the “zoom” interaction made more sense as a default for text, since a structure that is not scaled to the data often requires a user to navigate through extraneous nodes to find useful data. Guided by DG2, we chose these defaults per-modality according to each modality’s affordances. We also considered cases where the modalities are used together — for instance, a sighted collaborator brushing on the visualization to momentarily draw a screen reader user collaborator’s attention to a subset of data. However, our choice of default potentially trades off consistency across modalities — an important consideration for DG3.

Future work on interaction design for multi-modal data representations can work towards a better understanding of what types of approaches are best suited for certain situations or tasks, and how an author or end user might be able to switch between interaction techniques. And, though we conducted this initial exploration in the textual modality, future work remains to explore how interaction concepts like conditional encoding and viewport scaling extend to other non-visual modalities, like interactive sonification, in the context of a multi-modal system.

5.3.2 Preserving Interactive Context Across Modalities. Because the representations are designed to be used together, we wanted to enable users to smoothly switch between modalities to facilitate complementary use (DG2). This required us to think about how to maintain context when switching representations. In our initial explorations, a co-author compared the ability to select data via navigation in the textual structure with “pointing at part of a chart.”

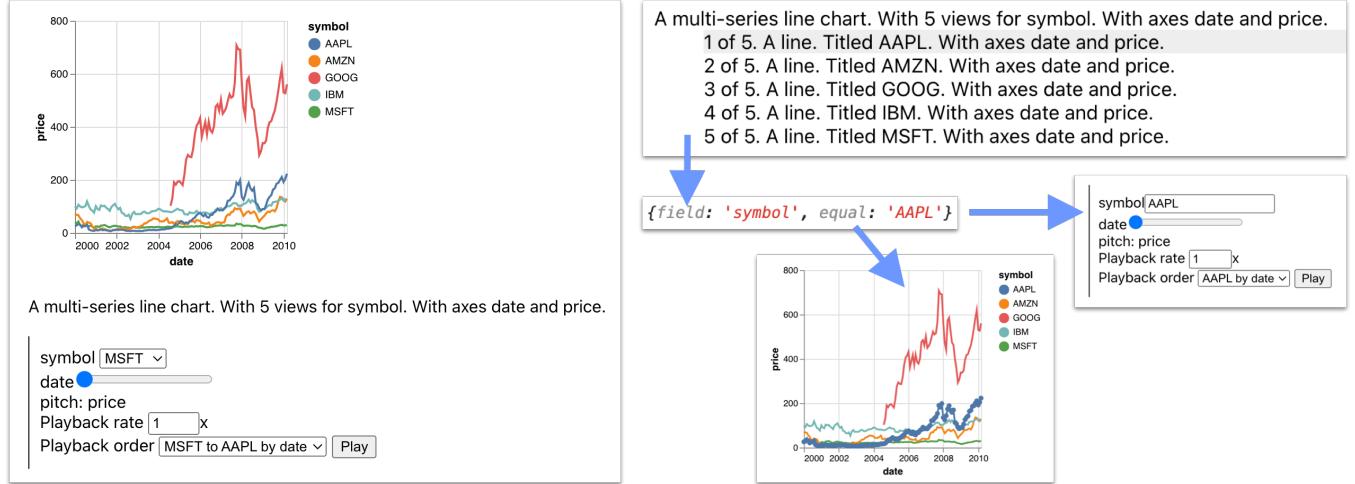


Figure 6: An example of linked interaction across modalities, driven by the textual modality. Navigating to a node in the textual structure emits a query predicate. The visualization reifies this predicate as a conditional encoding, and the sonification reifies it as a filter.

We designed linked interactions so that the system could express a consistent understanding of what data the user is “pointing” at across all representations.

Another important goal of linked interaction was to establish common ground for collaboration and presentation (DG3). One of the most important uses of data is to communicate with others, and not everyone in a conversation may use the same sensory modalities. This is why, despite primarily designing Umwelt with screen readers in mind, we found it important to include a visual representation that visually conveys the state of a screen reader user’s exploration in the textual structure or sonification. Conversely, the textual structure and sonification update to reflect interactions on the visualization. This also helps users think of the representations as complementary, e.g. by using one for wayfinding and the other for consuming [22] (DG2).

6 EVALUATION: EXAMPLE GALLERY

To evaluate our approach’s expressive extent, we used Umwelt to create a gallery of diverse multi-modal examples representing a variety of key-value semantics (Figure 7). In addition to simple examples consisting of one visualization, one audio track, and one textual structure (A, G), Umwelt provides a simple view composition abstraction that can express concatenated and layered visualizations (D, E), concatenated sonifications (B, C, D, E), and multi-view textual tree structures (A, D, E, F).

In contrast to prior approaches, such as Chart Reader [37], that were constrained to a small set of chart forms (and therefore key-value semantics), Umwelt can express more complex relationships among fields. For instance, Figure 7.B shows a dataset with an empty key and a set of values with two quantitative fields and one nominal field, visually represented as scatterplot. Because there are two quantitative values, a user could plausibly want to look up either one by pitch. Figure 7.B provides two audio units so that users can choose which value field to sonify. However, since there

is no key by which to look either value up, Figure 7.B’s sonification uses binning and aggregation to transform the fields, creating a 1:1 correspondence between bins and aggregated values so that users can traverse the bins to look up a value. The result is a 2 dimensional sonification that conveys the distribution of quantitative values in the x or y orientations.

Though this is not the only possible way to sonify a scatterplot, Figure 7.B is illustrative of the importance of decoupling visual and non-visual specification (DG1) in order to express representations that achieve complementary goals (DG2). A system that derives audio encodings from visual encodings might re-map the x and y encodings directly to pitch. A user might want to bin before sonification, as shown in Figure 7.B, in order to get a high level sonic overview of how the data is distributed along an axis without being overwhelmed by the fluctuating values of each individual data point. But in a visualization-first system, because the visualization is not binned, a user would not be able to add binning to the sonification without first changing the visualization to a binned representation (e.g. heatmap). Yet, the user may not want to align the modalities in this way; they may want to switch between the sonification overview and a visual or textual representation of individual data points. Umwelt’s approach enables a user to choose the set of representations that best suits their goals.

7 EVALUATION: USER STUDY

To evaluate Umwelt, we conducted remote studies with 5 expert BLV participants. Each participant met with us for two 90-minute sessions over Zoom video calling with screenshare for a total of 3 hours per participant. We split studies into two sessions to limit participant fatigue, and to give participants adequate time to become familiar enough with the system and its concepts to surface insights that reflect ordinary use conditions. In the first session, participants used the viewer to analyze an example dataset using multiple modalities. In the second session, participants used the

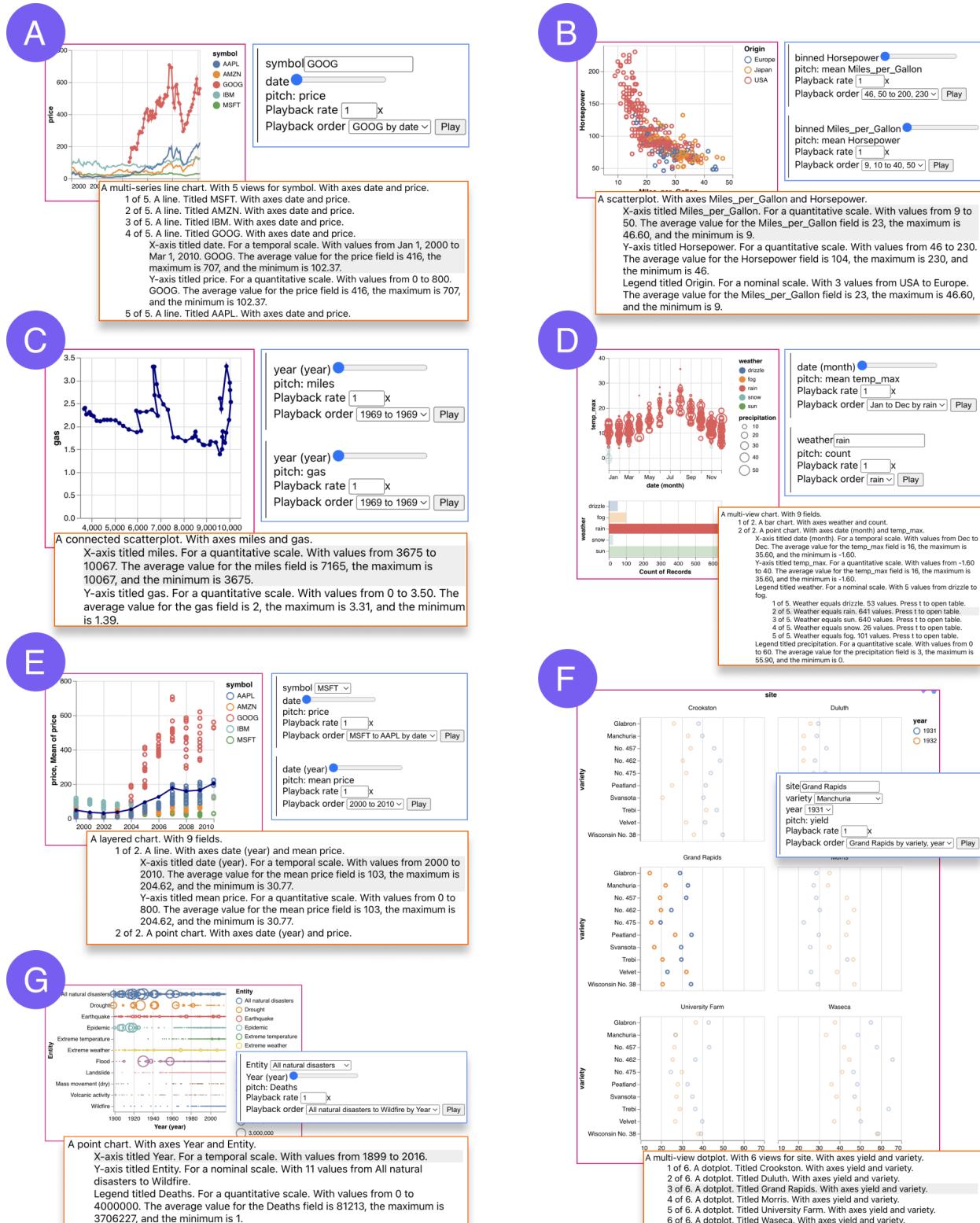


Figure 7: Example multi-modal representations created with Umwelt, expressing a range of key-value semantics. **A)** Two keys and one value. **B)** No keys and three values. Audio units represent the quantitative values' distribution in 2 dimensions. **C)** One key and two values. Audio units represent looking up each value with the same key. **D)** Concatenated visualization and sonification. **E)** Layered visualization with concatenated sonification. **F)** Three keys and one value. **G)** An alternate visual representation for the same key-value semantics as A.

structured editor to choose from a set of example datasets and create their own multi-modal representations. Participants were compensated \$250 for 3 hours. The purpose of the evaluation was primarily exploratory, seeking to form an initial understanding of how screen reader users approach multi-modal representations and their specification.

Because Umwelt is a tool designed with expert users in mind, we made an intentional choice to recruit a smaller number of participants to spend more time going in-depth with each participant. In qualitative research, the goal of selecting each qualitative case to examine is not to systematically answer descriptive questions about a population; it is to “ask how or why questions about processes unknown before the start of the study” [35]. Consequently, the goal of recruitment in our study was not to create the largest, most representative sample of a population, but to draw on participants’ lived experiences to reach a “saturation” of insights – building our understanding to a point at which adding more participants stops giving us new or surprising information [35]. We found that three hours per participant with five participants enabled us to reach saturation.

Recognizing that there is a history of exploitative relationships between researchers and marginalized research participants [24, 39], we reference our participants by name throughout the paper with their consent. As scholars of citational justice in HCI note, a lack of intentional decision-making about who to acknowledge for their intellectual contributions can lead to the erasure of marginalized individuals’ work and knowledge [23]. Our intention with this choice is to appropriately acknowledge and credit the expertise for which we recruited. Following methodological recommendations to be specific about our target population and concept of expertise [8], our study conceives of expertise along two dimensions: screen reader experience, and data analysis experience. We describe our participants’ backgrounds in Table 2.

7.1 Quantitative Results

We designed two Likert surveys to separately evaluate the user experience of the viewer and the editor. Participants responded on a scale of 1 to 5, where a higher number corresponds to an easier or more enjoyable experience. We report participants’ responses in Table 3. The median scores suggest that participants generally found both the viewer and editor fairly easy to learn and enjoyable to interact with. According to participants, the viewer facilitates trend and pattern exploration in the data, and transitioning between modalities within the viewer is straightforward. While participants rated the editor as slightly more difficult to learn, many also expressed interest in investing more time to learn because of its capabilities. When it comes to making edits, participants found the sonification settings easy to customize, but had a harder time predicting updates in the viewer based on the changes made in the editor. In the qualitative analysis section, we will further contextualize participants’ ratings.

7.2 Qualitative Results: Multi-Modal Viewer

7.2.1 Modalities have complementary affordances. Participants found it useful to have multiple modalities available for many reasons,

including increased optionality, modality-specific affordances, complementary uses leading to better understanding, and toggling between overview and detail.

Multiple representations as options to accommodate varying needs. Offering multiple modalities can help avoid cognitive or sensory overload. Minimizing cognitive load is a foundational principle in HCI; however, as research on accessible data analysis has shown, cognitive load can pose amplified challenges when it intersects with various disabilities [12]. As Erven noted, using only textual or tabular representations can result in “number fatigue” where the numbers “stop meaning anything.” This fatigue can be compounded for people with disabilities related to attention management or memory. Having the option to switch from textual representations to sonification can potentially help provide more usable alternatives.

For users who may need to commit additional effort to use certain representations, having alternatives can also help manage sensory fatigue. Perry, who works with low-vision colleagues, suggested that they might like to “rest [their] eyes [while] flipping through the data.” In these situations, being able to switch to a different representation can better accommodate an individual’s needs by balancing their sensory load.

Complementary modalities enable better understanding via overview and detail. Just like sighted visualization users, studies [32, 40] have shown that BLV users follow the information-seeking heuristic of “overview first, zoom and filter, and details on demand” [34]. As Bower noted, “when people look at a graph, they look at the big picture first and then they start scrutinizing it.” Participants found that sonification and textual description complement each other by effectively conveying overview and detail, respectively. Mustill-Rose noted that the textual description gave him the min, max, and average values, which are “hard if not impossible to get from sonification.” On the other hand, Perry enjoyed the ability to sonify “trend lines in the data without having to go point by point.” Since modalities afford different kinds of information-seeking operations, participants sought to choose the right modality for the task at hand. Switching between representations also helped participants adjust their initial assumptions about the data. For instance, Mustill-Rose listened to the sonification first and initially hypothesized that the stocks dataset contained only one data point per year. Then, he noticed that this was not the case when he explored the textual representation. He reflected that “the lesson there is to not consider just one modality.”

7.2.2 Synchronized query predicates help users share context between modalities. Participants valued the ability to maintain a shared query predicate while switching between modalities, which crucially helped them think of the modalities as different ways of looking into the same underlying data. Mustill-Rose described the synchronization across modalities as an “enabler” in the sense that “it’s decreasing the time that it’s taking me to get the data [from] the [time] period that I need” before he then “switch[es] to something else to look at it in a different way.” Because the system maintained his interactive context as he switched representations, he was able to stay in the flow of his ongoing analysis. This echoes prior findings that delays caused by interactive latency during data analysis

Table 2: Participants' names, demographic information, and descriptions of their screen reader and data analysis experience. Participants are referenced by name with their consent.

Name	Age Bracket	Self-Description of Disability	Screen Reader Experience	Data Analysis Experience
Ben Mustill-Rose	20-35	Totally blind, lost sight in early teens	Proficient with NVDA and sonification add-ons	Frequently analyzes data in Python as software engineer
Ken Perry	50+	100% blind, lost sight in early 20s	Proficient with JAWS and other screen readers	Teaches Python and other programming languages, has written statistical software
Dorene Cornwell	50+	Low vision / high partial, had detached retinas in mid-life	Proficient with JAWS screen reader + ZoomText for magnification	Masters-level courses in statistics and related fields
Liam Erven	35-50	No usable vision, hearing impaired in right ear	Teaches students how to use assistive technology, including all major screen readers	Basic proficiency, uses spreadsheets
Amy Bower	50+	Low partial vision, cannot see computer screen. Declining vision since mid-20s	Proficient in JAWS (self-taught)	Does research on oceanographic data. Uses Matlab for data analysis.

Table 3: Rating scores for the viewer and editor on a five-point Likert scale where 1 = Very Difficult (Very Unenjoyable) and 5 = Very Easy (Very Enjoyable). Median scores are shown in bold, averages in brackets [], standard deviations in parentheses ()

Viewer	Score	Editor	Score
How easy was it to learn to use the viewer?	4 [4] (0.71)	How easy was it to learn to use the editor?	3 [3.4] (0.55)
After understanding how the viewer works, how enjoyable was it to interact with the data?	5 [4.8] (0.45)	After understanding how the editor works, how enjoyable was it to edit the data representation?	4 [3.8] (0.45)
After understanding how the viewer works, how easy was it to switch between descriptions and sonifications on-demand?	4 [4.2] (0.45)	If you had a change you wanted to make to the data representation, how easy is it to understand how to make that change using the editor?	4 [3.8] (0.84)
How easy was it to be able to customize the sonification settings, including playback mode, audio axis ticks, and playback rate?	5 [4.6] (0.55)	After understanding how the editor works, how easy was it to predict how changes in the editor would affect the viewer?	3 [3.6] (0.89)
After understanding how the viewer works, how easy was it to understand trends and patterns in the data?	4 [4.4] (0.55)	After understanding how the editor works, how easy was it to check the result of your edits in the viewer?	4 [4] (0.71)

can “[disrupt] fluent interaction” and cause people to lose their train of thought during exploratory analysis [25].

7.2.3 Customization supports differences in task and experience. Research has shown that customizable textual descriptions support users who have different preferences or tasks, allowing them to control the information they receive and how it's presented [22]. This was reiterated by Perry, Cornwell, and Erven, who encountered situations where they wanted to adjust the presence, verbosity, and ordering of information in text. We also found this customizability idea applicable beyond textual description, particularly for Jones et al.'s wayfinding and consuming affordances [22].

Wayfinding. The audio axis ticks feature supports wayfinding by helping users understand their progress through a temporally proceeding sonification. However, it trades off efficiency, and becomes less necessary over time as users get more familiar with the data. Mustill-Rose found himself wanting to disable the axis ticks after listening to a few sonifications. He said, “at first it was useful [...] but now that I know what I'm looking at, I feel like the [audio axis ticks] has proved its value. And now I don't need it anymore.” However, once he selected a different subset of the data, he realized that it was “now useful again because I haven't explored this section.” His need for this feature was situational throughout his analysis, depending on whether he was focusing on learning

the layout of the data or “understanding and honing in on” the data. As a result, the ability to enable or disable the axis ticks was important to offer as a customization.

Consuming. Another important customization was the sonification’s playback speed. As Cornwell noted, preferred screen reader reading speed varies widely among BLV individuals. For sonification, participants considered their base preference as well as their familiarity and task-specific needs. Perry and Bower both noted that their preferred playback speed was situational. Perry noted that he “would get used to it faster, but [he] would start slower because it gives more time to listen to each point.” Slower speeds were better when initially learning about the data, and he would speed up as he became more familiar.

7.2.4 Multi-modal representations facilitate communication between people who rely on different senses. In a multi-modal system, participants who were not primarily using the visualization still valued the presence of synchronized visual highlighting and references to visual concepts in the description. As BLV professionals who work with sighted colleagues, participants frequently encounter situations where they need to establish common ground with others who primarily use different senses.

Contributing confidently to data-driven discussions. As a software engineer who works with only sighted colleagues, Mustill-Rose stressed that an important goal of data analysis is to have enough information to “participate meaningfully in a discussion.” At minimum, he said, he wanted to be in Zoom meetings and “not seem clueless,” because as the only blind person on a team, consistently being the only person who can’t comment on a topic can compound with unconscious bias to affect promotions and work opportunities. Erven echoed this sentiment, saying that “the most important thing is independence.” Visual modalities are only helpful when “it’s not something you need to rely on to do your work,” forcing BLV users to rely on sighted help. Instead, as previous work has also argued [40], accessible representations should promote user agency for self-guided analysis — and for BLV people not only to participate in, but also create and lead data-driven discussions.

Presenting to mixed audiences. In her job, Cornwell frequently makes presentations to majority sighted audiences. As a result, “visual charts are always useful” to her. She explained, “if I’m needing to talk about [the data], I can just say, look at the red line and the people who are really visual — that’s an important source of interactivity for them.” Additionally, synchronization between modalities plays a helpful role in presentation. For example, Cornwell imagined a hypothetical situation where she played a sonification while presenting, and sighted audience members followed along on the visualization. In this situation, having multiple modalities would make the presentation more accessible and also help communicate the data more effectively.

Collaborating across different levels of vision. Many participants frequently collaborate directly with others with different levels of vision. Cornwell mentioned working with someone who was totally blind, and thought that “sonification on a screen share” would be extremely valuable for communicating about data. Similarly, Bower felt that the visual highlighting of her selection in the textual structure and sonification would help a collaborator “get on the same page” and help them “know where [she’s] looking.” She

drew an analogy to pointing at something on a visual chart, as a way of directing a collaborator’s attention.

7.3 Qualitative Results: Structured Editor

7.3.1 Users want, but lack, interfaces for creating data representations. Participants have existing strategies for working with data that primarily involve spreadsheets and scripting. Erven, Perry, Cornwell, and Bower reported using Excel or Google Sheets; Mustill-Rose, Perry, and Bower reported writing their own scripts in various tools, including python, octave, and matlab. However, there was consensus that these existing workflows are insufficient. Cornwell put it succinctly when describing raw data: “no one wants to read that stuff.” But with the exception of Bower (who has used Highcharts Sonification Studio and SAS Graphics Accelerator), no participants could think of available tools for creating their own representations without having to write code.

End-user tools are important because they lower the technical barrier for creating representations. When comparing the Umwelt editor to writing code, Mustill-Rose said that “there’s less cognitive pressure using a UI than if I was having to write code to do it.” However, sometimes tools can overly complicate the process of making a simple representation. Bower said, “I don’t care about instruments and timbre and all that, I just want access to a time series.” Because of high up-front specification cost, some tools are too difficult to use for even simple cases. Nonetheless, Bower is interested in trying new tools for creating data representations, saying, “I’m kind of desperate for anything” that’s usable and accessible.

7.3.2 Structured editing with default specifications reduces semantic and articulatory distance. When participants decide to create representations, they face challenges to do with semantic and articulatory distance [21]. In HCI theory, semantic distance is the distance between a user’s intentions and how these intentions are translated into the concepts provided by a user interface. Similarly, articulatory distance is the relationship between an interface’s concepts and the set of physical actions a user has to take to express something in terms of those concepts.

Semantic distance. When Perry approached analyzing the penguin dataset, he initially said, “I want to compare beak length, body mass, and sex altogether — I want to see the graph for all three of these together.” Though Perry knew that he wanted to specify visual encodings that would be reflected in the textual hierarchy, he did not immediately know what those encodings were. This was a problem of semantic distance, because he needed to map his goal onto the concepts provided by the user interface. Luckily, the heuristics generated a default specification for that set of fields that matched his expectations. As a result, he was still able to create the chart despite lack of familiarity with visualization concepts. However, when the system was not able to generate a default specification for Cornwell, she remarked that it was hard to figure out “which functions apply, like figuring out if I wanted it grouped by island or species.” Even though she had goals in mind, it was difficult for her to translate those goals into specific encodings and field transformations. This suggests a need for future work on bridging semantic distance — for instance, by designing high-level abstractions that adhere closer to users’ abstract goals, reducing the amount of translation work.

Articulatory distance. Using the editor, Mustill-Rose remarked that “if I were writing code, I’d need to think about what the end result was and what code I needed to write to achieve it at the same time.” Rather than having to remember the names of functions and expressions in a textual language, Mustill-Rose was able to use commonplace HTML input elements that express atomic edits to a specification as simple button clicks or dropdown selections. However, the editor also has limitations when it comes to articulatory distance. Cornwell, who had created a chart that was not a default specification, noted that a main challenge was that “when you’re looking at the fields, you have to add encodings for everything you want.” When specifying multi-modal representations, there can be a lot of repetitive operations to create three outputs that are conceptually similar.

7.3.3 Users think in both field-oriented and encoding-oriented terms. Throughout the specification process, we observed that participants moved between field-oriented and encoding-oriented ways of thinking. For most participants, the tendency was to begin by identifying a set of fields they were interested in. For instance, Erven commented that it felt natural to begin by “choosing the fields you want,” since you “might not want all that data.” When default specifications matched their expected representation, or when they only required minor edits, participants were generally content with the result that they achieved through field-oriented specification.

However, when more manual editing was required to achieve the desired output, we found that participants shifted more toward encoding-oriented specification as they envisioned specific output representations. Cornwell initially stated her goal by saying, “I want to know what species are on which island and then I want a sex distribution.” At this point in his process, she had not committed to any encoding properties or specific modalities, but was envisioning the semantics and structure of the data in terms of relationships between fields. After selecting the relevant fields, she began to add encodings, and then became somewhat stuck. When prompted to reiterate her goal, she said that she wanted to create a “bar chart with island on the x-axis and count for species for the y-axis.” At this point further into the process, she had imagined a specific visual representation, which she was attempting to decompose into encodings and then map onto editor operations.

Interestingly, Bower – who is familiar with both visualization and sonification – had a mental model that blurred the dichotomy between field- and encoding-oriented specification. She initially approached the Seattle weather dataset by selecting date and temp_max. When she tabbed down the editor to read the default specification, she noticed that Umwelt had assigned y and pitch encodings to temp_max. Based on her extensive previous experience with data visualization and sonification, she remarked, “I merge those in my head – I think of those as the same thing.” This suggests that even when thinking in encoding-oriented terms, Bower was reasoning about the data’s underlying key-value semantics.

8 DISCUSSION AND FUTURE WORK

We contribute Umwelt, an accessible authoring environment designed to de-center the visual modality in data analysis. Umwelt allows users to specify data representations, including visualization, structured textual description, and sonification, using a shared

abstract data model. Unlike existing tools, Umwelt does not rely on an existing visual specification, affording users more flexibility in prototyping multi-modal representations. The editor’s state is reflected in independent visual, textual, and sonification views linked through shared interactions, encouraging complementary use of multiple modalities. In this section, we discuss potential directions for future work surfaced by Umwelt.

8.1 Designing Natively Non-Visual Data Representations

Differences in a representation’s modality affect how information is presented to a user, and the operations the user needs to perform to access the information. For example, a screen reader must “explicitly linearize reading a visualization” in order to narrate elements one at a time [40] – in contrast to how visual perception enables a user to move around parts of a visualization relatively freely. Similarly, researchers have compared tactile perception to “reading a map through a small tube” [18, 40]. An implication of these modality differences, as participants in our study found, is that users find some modalities inherently more suited to certain tasks than others. Further, it suggests that due to medium-specificity, it is not always possible to directly translate a data representation from one modality to another while maintaining 1:1 support for the same set of tasks.

However, existing systems for authoring non-visual representations largely attempt to directly translate source visualizations into standalone non-visual replacements. For example, while Highcharts Sonification Studio [9] successfully translates single-series line charts into equivalent sonifications, this approach breaks down for scatterplots. This is because the way a sighted user reads a scatterplot has no unambiguous analogue in the medium of sonification, which imposes a linearized traversal order over the data. In contrast, Umwelt’s default specification heuristics pair scatterplots with sonifications that diverge from the visualization by adding additional binning and aggregation, in order to prioritize conveying the data’s 2d distribution.

In the context of multimodal representations, Umwelt advances the idea that a representation should prioritize fit with its modality’s affordances over fidelity to the visual representation. This conceptual orientation has implications for the design of future non-visual representations. For example, current approaches to tactile charts largely focus on converting visual channels to tactile ones while otherwise faithfully recreating the visualization [13]. Instead, future work could explore tactile-first designs that make more intentional use of the processual, part to whole [17] nature of tactile perception.

8.2 Interdependence and Relational Dimensions of Access

In designing Umwelt, we advocate for a conceptual shift in the field of accessible data visualization – focusing not only on making existing visualization accessible to BLV readers, but also on empowering BLV data analysts to independently produce their own representations and conduct self-guided data exploration. Because existing approaches that center the visual modality can sometimes

create barriers or reinforce BLV users' dependence on sighted assistance, we believe a focus on independence to be an important step forward. However, in addition to independence, disability scholars have advanced *interdependence* as a complementary conceptual frame [5]. An interdependence frame acknowledges that all people constantly depend on others, and so a focus on relationships is necessary to understand how access is socially created in practice.

Our initial evaluation of Umwelt surfaced ideas that suggest the need for interdependence (alongside independence) as a lens for design. For instance, participants felt that building common ground between mixed ability colleagues in workplace settings was important to their career advancement, highlighting the fact that BLV people's access needs are embedded in a social and relational context. Future work can motivate and inspire system design based on not only how BLV users can get immediate access to information in data, but also how they hope to use that information to participate in broader social processes.

ACKNOWLEDGMENTS

Thank you to Josh Pollock, Amy Fox, and Shuli Jones. This work was supported by NSF #1942659 and #1900991, and the MIT Morningside Academy for Design Fellowship.

REFERENCES

- [1] 2017. SAS Graphics Accelerator. <https://support.sas.com/software/products/graphics-accelerator/index.html>
- [2] Alice Wong and Ed Yong. 2022. What Counts as Seeing. *Orion Magazine* (2022). <https://orionmagazine.org/article/ed-yong-alice-wong-interview/>
- [3] Catherine M. Baker, Lauren R. Milne, Ryan Drapeau, Jeffrey Scofield, Cynthia L. Bennett, and Richard E. Ladner. 2016. Tactile Graphics with a Voice. *ACM Transactions on Accessible Computing* 8, 1 (Jan. 2016), 3:1–3:22. <https://doi.org/10.1145/2854005>
- [4] Catherine M. Baker, Lauren R. Milne, Jeffrey Scofield, Cynthia L. Bennett, and Richard E. Ladner. 2014. Tactile graphics with a voice: using QR codes to access text in tactile graphics. In *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility (ASSETS '14)*. Association for Computing Machinery, New York, NY, USA, 75–82. <https://doi.org/10.1145/2661334.2661366>
- [5] Cynthia L. Bennett, Erin Brady, and Stacy M. Brannah. 2018. Interdependence as a Frame for Assistive Technology Research and Design. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, Galway Ireland, 161–173. <https://doi.org/10.1145/3234695.3236348>
- [6] A. F. Blackwell, C. Britton, A. Cox, T. R. G. Green, C. Gurr, G. Kadoda, M. S. Kutar, M. Loomes, C. L. Nehaniv, M. Petre, C. Roast, C. Roe, A. Wong, and R. M. Young. 2001. Cognitive Dimensions of Notations: Design Tools for Cognitive Technology. In *Cognitive Technology: Instruments of Mind (Lecture Notes in Computer Science)*. Meurig Beynon, Christopher L. Nehaniv, and Kerstin Dautenhahn (Eds.). Springer, Berlin, Heidelberg, 325–341. https://doi.org/10.1007/3-540-44617-6_31
- [7] Matthew 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] Alyxander Burns, Christiana Lee, Ria Chawla, Evan Peck, and Narges Mahyar. 2023. Who Do We Mean When We Talk About Visualization Novices?. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–16. <https://doi.org/10.1145/3544548.3581524>
- [9] Stanley J. Cantrell, Bruce N. Walker, and Øystein Moseng. 2021. Highcharts Sonification Studio: An Online, Open-Source, Extensible, and Accessible Data Sonification Tool. In *Proceedings of the 26th International Conference on Auditory Display (ICAD 2021)*. International Community for Auditory Display, Virtual Conference, 210–216. <https://doi.org/10.21785/icad2021.005>
- [10] E F Codd. 1971. *Further Normalization of the Data Base Relational Model*. Technical Report RJ909. IBM, San Jose, California.
- [11] Scott Davert and AppleVis Editorial Team. 2019. What's New In iOS 13 Accessibility For Individuals Who Are Blind or Deaf-Blind. <https://www.applevis.com/blog/whats-new-ios-13-accessibility-individuals-who-are-blind-or-deaf-blind>
- [12] Doug Schepers. 2022. Designing Data for Cognitive Load. In *Centering Accessibility in Data Visualization*, Jonathan Schwabish, Sue Popkin, and Alice Feng (Eds.). Urban Institute. <https://www.urban.org/sites/default/files/2022-12/Do%20No%20Harm%20Guide%20Centering%20Accessibility%20in%20Data%20Visualization.pdf>
- [13] Christin Engel, Emma Franziska Müller, and Gerhard Weber. 2019. SVGPLIT: an accessible tool to generate highly adaptable, accessible audio-tactile charts for and from blind and visually impaired people. In *Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments (PETRA '19)*. Association for Computing Machinery, New York, NY, USA, 186–195. <https://doi.org/10.1145/3316782.3316793>
- [14] Danyang Fan, Alexa Fay Siu, Hrishikesh Rao, Gene Sung-Ho Kim, Xavier Vazquez, Lucy Greco, Sile O'Modhrain, and Sean Follmer. 2023. The Accessibility of Data Visualizations on the Web for Screen Reader Users: Practices and Experiences During COVID-19. *ACM Transactions on Accessible Computing* 16, 1 (March 2023), 4:1–4:29. <https://doi.org/10.1145/3557899>
- [15] Danyang Fan, Alexa Fay Siu, Wing-Sum Adrienne Law, Raymond Ruihong Zhen, Sile O'Modhrain, and Sean Follmer. 2022. Slide-Tone and Tilt-Tone: 1-DOF Haptic Techniques for Conveying Shape Characteristics of Graphs to Blind Users. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–19. <https://doi.org/10.1145/3491102.3517790>
- [16] A. Jonathan R. Godfrey and M. Theodor Loots. 2015. Advice from Blind Teachers on How to Teach Statistics to Blind Students. *Journal of Statistics Education* 23, 3 (2015). Publisher: American Statistical Association ERIC Number: EJ1083113.
- [17] Lucia Hasty. 2017. Teaching Tactile Graphics. <http://www.perkinselearning.org/videos/webcast/teaching-tactile-graphics>
- [18] Lucia Hasty, Janet Milbury, Irene Miller, Allison O'Day, Pather Aquinas, and Diance Spence. 2011. *Guidelines and Standards for Tactile Graphics*. Technical Report. Braille Authority of North America. <http://www.brailleauthority.org/tg/>
- [19] Leona M Holloway, Cagatay Goncu, Alon Ilsar, Matthew Butler, and Kim Marriott. 2022. Infosonics: Accessible Infographics for People who are Blind using Sonification and Voice. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3491102.3517465>
- [20] Amy Hurst and Jasmine Tobias. 2011. Empowering individuals with do-it-yourself assistive technology. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility (ASSETS '11)*. Association for Computing Machinery, New York, NY, USA, 11–18. <https://doi.org/10.1145/2049536.2049541>
- [21] Edwin L. Hutchins, James D Hollan, and Donald A Norman. 1985. Direct manipulation interfaces. *Human-Computer Interaction* 1 (1985), 311–338.
- [22] Shuli Jones, Isabella Pedraza Pineros, Daniel Hajas, Jonathan Zong, and Arvind Satyanarayan. 2023. "Customization is Key": Four Characteristics of Textual Affordances for Accessible Data Visualization. <https://doi.org/10.48550/arXiv.2307.08773> arXiv:2307.08773 [cs].
- [23] Neha Kumar and Naveena Karusala. 2021. Braving Citational Justice in Human-Computer Interaction. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems (CHI EA '21)*. Association for Computing Machinery, New York, NY, USA, 1–9. <https://doi.org/10.1145/3411763.3450389>
- [24] Calvin A. Liang, Sean A. Munson, and Julie A. Kientz. 2021. Embracing Four Tensions in Human-Computer Interaction Research with Marginalized People. *ACM Transactions on Computer-Human Interaction* 28, 2 (April 2021), 14:1–14:47. <https://doi.org/10.1145/3443686>
- [25] Zhicheng Liu, Biye Jiang, and Jeffrey Heer. 2013. imMens: Real-time Visual Querying of Big Data. *Computer Graphics Forum* 32, 3pt4 (June 2013), 421–430. <https://doi.org/10.1111/cgf.12129>
- [26] Tamara Munzner. 2014. *Visualization Analysis and Design*. A K Peters/CRC Press, New York. <https://doi.org/10.1201/b17511>
- [27] Venkatesh Potluri, John Thompson, James Devine, Bongshin Lee, Nora Morsi, Peli De Halleux, Steve Hodges, and Jennifer Mankoff. 2022. PSST: Enabling Blind or Visually Impaired Developers to Author Sonifications of Streaming Sensor Data. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22)*. Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3526113.3545700>
- [28] Donghao Ren, Bongshin Lee, Matthew Brehmer, and Nathalie Henry Riche. 2018. Reflecting on the Evaluation of Visualization Authoring Systems : Position Paper. In *2018 IEEE Evaluation and Beyond - Methodological Approaches for Visualization (BELIV)*, 86–92. <https://doi.org/10.1109/BELIV.2018.8634297>
- [29] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko. 2008. Effectiveness of Animation in Trend Visualization. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (Nov. 2008), 1325–1332. <https://doi.org/10.1109/TVCG.2008.125>
- [30] Hans Rosling. 2006. The best stats you've ever seen. https://www.ted.com/talks/hans_rosling_the_best_stats_you_ve_ever_seen https://www.ted.com/talks/hans_rosling_the_best_stats_you_ve_ever_seen
- [31] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2017. Vega-Lite: A Grammar of Interactive Graphics. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (Jan. 2017), 341–350. <https://doi.org/10.1109/TVCG.2016.2599030>
- [32] Ather Sharif, Sanjana Shivani Chintalapati, Jacob O. Wobbrock, and Katharina Reinecke. 2021. Understanding Screen-Reader Users' Experiences with Online Data

Visualizations. In *ACM Conference on Computers and Accessibility (SIGACCESS '21)*. New York, NY, USA, 1–16. <https://doi.org/10.1145/3441852.3471202>

[33] Ather Sharif, Olivia H. Wang, Alida T. Muongchan, Katharina Reinecke, and Jacob O. Wobbrock. 2022. VoxLens: Making Online Data Visualizations Accessible with an Interactive JavaScript Plug-In. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–19. <https://doi.org/10.1145/3491102.3517431>

[34] Ben Shneiderman. 2003. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *The Craft of Information Visualization*, BENJAMIN B. Bederson and BEN Shneiderman (Eds.). Morgan Kaufmann, San Francisco, 364–371. <https://doi.org/10.1016/B978-155860915-0/50046-9>

[35] Mario Luis Small. 2009. 'How many cases do I need?': On science and the logic of case selection in field-based research. *Ethnography* 10, 1 (March 2009), 5–38. <https://doi.org/10.1177/1466138108099586> Publisher: SAGE Publications.

[36] Chris Stolte, Diane Tang, and Pat Hanrahan. 2002. Polaris: A System for Query, Analysis, and Visualization of Multidimensional Relational Databases. *IEEE*

Transactions on Visualization and Computer Graphics 8, 1 (2002).

[37] 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 (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–18. <https://doi.org/10.1145/3544548.3581186>

[38] Jakob von Uexküll. 2010. *A foray into the worlds of animals and humans: with a theory of meaning*. University of Minnesota Press, Minneapolis.

[39] Anon Ymous, Katta Spiel, Os Keyes, Rua M. Williams, Judith Good, Eva Hornecker, and Cynthia L. Bennett. 2020. "I am just terrified of my future" Epistemic Violence in Disability Related Technology Research. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–16. <https://doi.org/10.1145/3334480.3381828>

[40] 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* (2022). <https://doi.org/10.1111/cgf.14519>