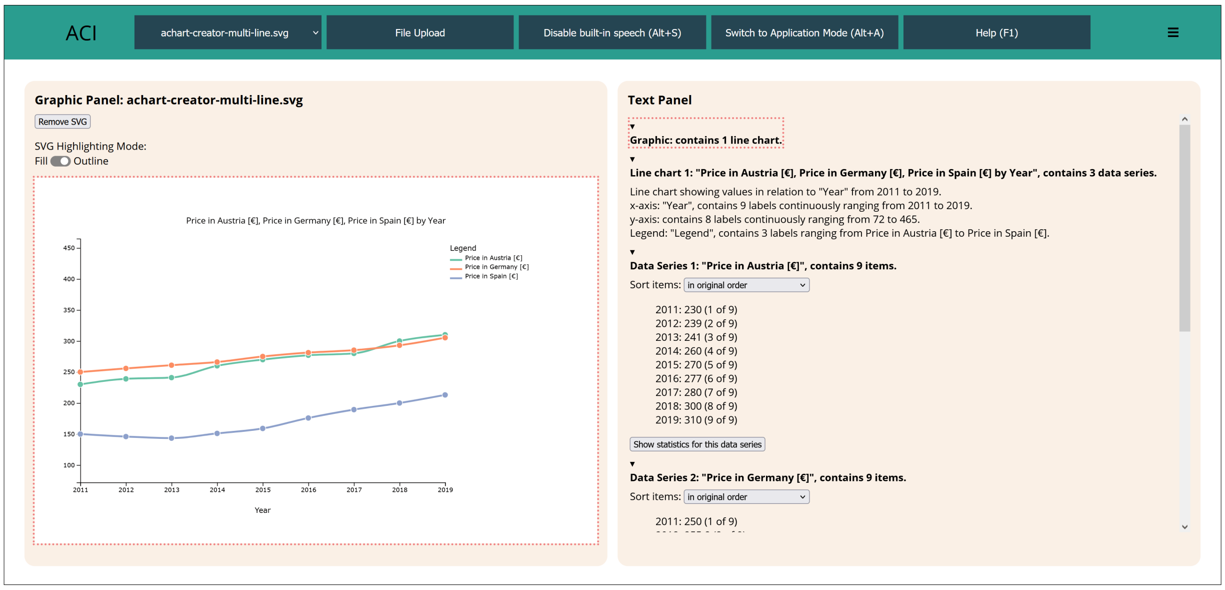
Accessible SVG Charts with AChart

Keith Andrews[*]
Graz University of Technology

Christopher Alexander Kopel[†]
Graz University of Technology


ACI
achart-creator-multi-line.svg
File Upload
Disable built-in speech (Alt+S)
Switch to Application Mode (Alt+A)
Help (F1)
Remove SVG
Price in Austria [€], Price in Germany [€], Price in Spain [€] by Year
450
400
350
300
250
200
150
100
2011
2012
2013
2014
2015
2016
2017
2018
2019
Year
Legend
Price in Austria [€]
Price in Germany [€]
Price in Spain [€]
Show statistics for this data series

Graphic Panel: achart-creator-multi-line.svg

SVG Highlighting Mode:
Fill Outline

Text Panel

▾
Graphic: contains 1 line chart.

▾
Line chart 1: "Price in Austria [€], Price in Germany [€], Price in Spain [€] by Year", contains 3 data series.

Line chart showing values in relation to "Year" from 2011 to 2019.
x-axis: "Year", contains 9 labels continuously ranging from 2011 to 2019.
y-axis: contains 8 labels continuously ranging from 72 to 465.
Legend: "Legend", contains 3 labels ranging from Price in Austria [€] to Price in Spain [€].

▾
Data Series 1: "Price in Austria [€]", contains 9 items.

Sort items: in original order

2011: 230 (1 of 9)
2012: 239 (2 of 9)
2013: 241 (3 of 9)
2014: 260 (4 of 9)
2015: 270 (5 of 9)
2016: 277 (6 of 9)
2017: 280 (7 of 9)
2018: 300 (8 of 9)
2019: 310 (9 of 9)

▾
Data Series 2: "Price in Germany [€]", contains 9 items.

Sort items: in original order

2011: 250 (1 of 9)

Figure 1: AChart Interpreter showing an accessible multi-line chart.

Abstract

AChart is a suite of open-source web-based tools written in TypeScript with Node.js to create and interpret semantically-enriched SVG-based accessible charts. AChart Creator is a command-line tool which generates accessible SVG charts from CSV files using the D3 framework, by injecting ARIA roles and properties from the AChart taxonomy. AChart Interpreter is a client-side web application and executable package which interprets such a semantically-enriched SVG chart and displays side-by-side graphical and textual versions of the chart. It can read out the chart using synthetic speech and its user interface is screen reader compatible. It can be used both by blind users to gain an understanding of a chart, as well as by developers and chart authors to verify the accessibility markup of an SVG chart. AChart Summariser is a command-line tool which interprets an accessible SVG chart and outputs a textual summary of the chart. AChart currently supports bar charts, line charts, and pie charts.

Index Terms: Data visualisation, charts, accessibility, WAI-ARIA roles and properties for SVG, open-source, web-based, TypeScript, JavaScript, D3, speech output, screen reader.

1 Introduction

Interest in the provision of accessible forms of graphics, charts, and visualisations has been steadily growing, culminating in this 1st Workshop on Accessible Data Visualization at IEEE VIS 2024. The field of web accessibility has long sought to make web pages more accessible through the use of semantic HTML elements, alternative texts, WAI-ARIA roles, states, and properties [1], `tabindex` attributes, and keyboard event listeners.

Many approaches have been proposed to present graphics, charts, and visualisations in non-visual ways to make them more accessible. Physical, tactile representations have been created or 3d-printed, both static and refreshable to data changes. Speech and non-speech audio (sonification) have been used to present data acoustically. Screen reader friendly output can be navigated by keyboard and output as speech or sent to Braille displays.

2 Semantic Enrichment of SVG Charts

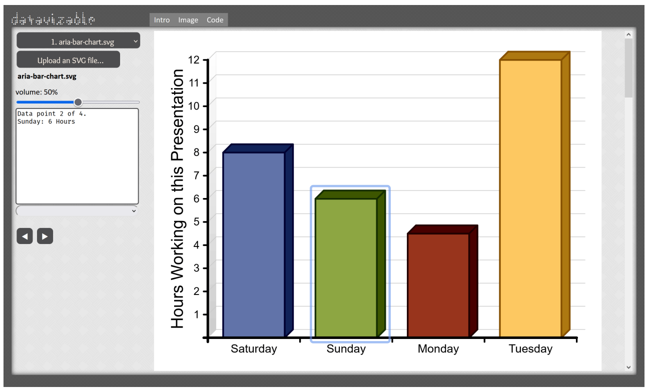
Data and semantics can be stored in machine-readable form within a Scalable Vector Graphics (SVG) chart using SVG text elements and additional annotations in the form of ARIA roles and properties and CSS class names. SVG provides three elements for including text inside a chart: `<text>` for visible text, `<title>` for accessible names (often rendered as tooltips on mouse-over), and `<desc>` for accessible descriptions (not rendered visually).

There are dozens of WAI-ARIA attributes (roles, properties, and states) which can add semantic annotations to elements on web pages:

- *Roles*: Indicate the meaning of an element, for example `role="button"`, `role="checkbox"`, or `role="article"`.
- *Properties*: Attach data and other information to an element, for example `aria-label="OK"`, `aria-labelledby="id-of-label-element"`, or `aria-haspopup="true"`.
- *States*: Indicate current state of an element, for example `aria-checked="false"` or `aria-pressed="true"`.

Initially intended for use with HTML elements, they can also be used with SVG elements.

[*]e-mail: kandrews@tugraz.at
[†]e-mail: chr.kopel@gmail.com



```
Data point 2 of 4.
Sunday: 6 Hours
```

Figure 2: Describler with an accessible bar chart.

Furthermore, the WAI-ARIA Graphics Module [2] defines three additional roles for graphics documents: `graphics-document` for a whole structured, navigable graphic, `graphics-object` for elements of graphical structure, and `graphics-symbol` for single objects whose concrete visual appearance is semantically irrelevant (like icons). Further custom roles and properties for charts and visualisations have been proposed, but are more experimental than standardised.

Describler is an experimental web application for browsing semantically enriched SVG charts and data visualisations [3, 4, 5]. Describler uses dedicated ARIA markup within a chart to convey its structure. A user can navigate within the chart by keyboard or mouse and elements of the chart are read out using speech synthesis. The navigation facilities and the speech synthesis work independently of any screen reader installed or running. The Describler application is shown in Figure 2.

3 Accessible Charts with AChart

The AChart (Accessible Chart) project was launched in 2019 with the goal of providing an open-source software solution for producing and interpreting accessible charts and data visualisations in Scalable Vector Graphics (SVG) format. AChart consists of two complementary software tools: AChart Creator generates semantically-enriched accessible SVG charts from tabular data, and AChart Interpreter interprets and reads out the accessible charts. A third tool, AChart Summariser, produces a solely textual summary of an accessible chart to the console.

Describler [4] served as an inspiring proof-of-concept. AChart's set of ARIA roles and properties extends and builds upon Describler's approach. AChart uses roles like `chart`, `chartarea`, `xaxis` , `yaxis`, `axislabel`, `datagroup`, `datapoint`, and `datavalue`. In addition, AChart uses properties like `aria-charttype` (with values `bar`, `line` or `pie`) and `aria-axistype`, as well as standard properties like `aria-valuemin`, `aria-valuemax`, and `aria-labelledby`. The full set of roles and properties used in AChart, and the differences to Describler, are described in Kopel [6, Chapter 5]. An example chart created by AChart can be seen in Figure 3 and Listing 1.

4 AChart Creator

AChart Creator is a command-line tool (`acreate`) which creates semantically-enriched accessible SVG charts. It currently supports bar charts, line charts, and pie charts. The tool reads tabular data from a CSV file and saves the resulting chart to an SVG file. The semantics are embedded using native SVG `<title>`, `<desc>`, and `<text>` elements, in combination with roles and properties from the AChart taxonomy applied to these and other SVG elements.

The tool is built in the Node.js environment and uses Nexe [7] to create binary executables. The JavaScript library D3 [8] is used to construct the charts. Since D3 assumes that it is being run inside a web browser, a simulated DOM is provided by the jsdom package [9].

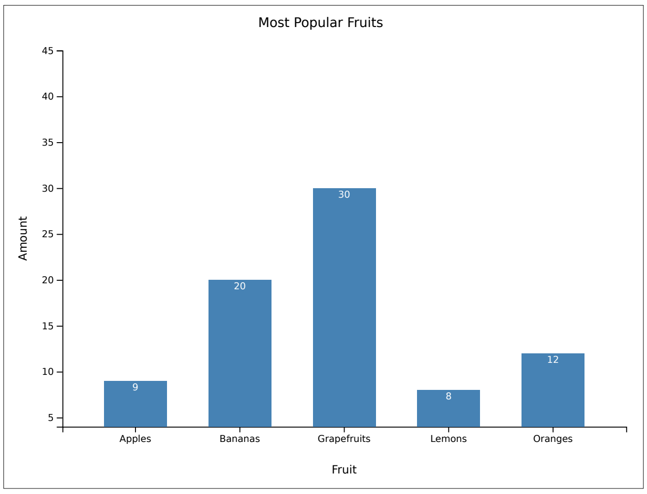


Figure 3: Accessible bar chart created with AChart Creator.

The syntax and options of the command can be seen in Listing 2. A dataset and example command are shown in Listings 3 and 4 respectively. These were used to create the chart shown in Listing 1 and Figure 3. AChart Creator is described in detail in Kopel [6, Chapter 6]. A showcase video was created by Perko [10]. The software is open-source and is available on GitHub [11].

5 AChart Interpreter

AChart Interpreter is a web application for interpreting semantically-enriched accessible charts in SVG format. It is a kind of screen reader for charts. The application analyses a chart to create an internal representation of relevant elements, then displays both the visual chart and a corresponding textual representation side by side, as can be seen in Figure 1. The Graphic Panel on the left shows the SVG chart, the Text Panel on the right shows the derived textual summary.

The current focus element is visually highlighted in both chart and textual summary and is read out. Navigation is possible both with keyboard (tab, return, and arrow key) and mouse. In Figure 4, the user has navigated to the third data point of Data Series 1. AChart Interpreter is screen reader compatible, but also has built-in speech output (using the Web Speech API [12]), in case no screen reader is available, or for use by sighted users. It can be used by developers and chart authors verify the accessibility markup, as well as by unsighted end users to explore a chart and have it read out.

The application is written in TypeScript [13], and can be deployed as a web application. In addition, Electron is used to build self-contained, standalone binary executable packages for various platforms [14]. AChart Interpreter is described in detail in Kopel [6, Chapter 7]. A showcase video was created by Perko [15]. The software is open-source and is available on GitHub [16]. A live demo is also available [17].

6 AChart Summarizer

AChart Summariser is a command-line tool, a spin-off of AChart Interpreter, which outputs a textual summary of an accessible SVG chart as plain text. With the tool, it is possible to perform automatic sequential analysis of multiple charts using shell scripts and programmatically process the output.

```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1"
  viewBox="0 0 750 600" role="graphics-document">
  ...
  <g id="ChartRoot" role="chart" tabindex="0" transform="translate(100,100)"
    aria-labelledby="title desc" aria-charttype="bar" aria-roledescription="Bar Chart">
    <desc id="desc">Each fruit with volume sold.</desc>
    <rect role="chartarea" width="600" height="400" fill="none"/>
    <text id="title" role="heading" text-anchor="middle"
      font-size="14" x="275" y="-25">Most Popular Fruits</text>

    <g id="xScale" role="xaxis" aria-axistype="category" aria-roledescription="x-Axis"
      aria-labelledby="x-title" tabindex="0" transform="translate(0,400)" ... >
      <text y="50" x="300" text-anchor="middle" fill="black" font-size="12"
        role="heading" id="x-title">Fruit</text>
      <path class="domain" stroke="currentColor" d="M0.5,6V0.5H600.5V6"/>
      <g class="tick" opacity="1" transform="translate(77.778,0)">
        <line stroke="currentColor" y2="6"/>
        <text fill="currentColor" y="9" dy="0.71em" role="axislabel" id="x1">Apples</text>
      </g>
      <g class="tick" opacity="1" transform="translate(188.889,0)">
        <line stroke="currentColor" y2="6"/>
        <text fill="currentColor" y="9" dy="0.71em" role="axislabel" id="x2">Bananas</text>
      </g>
      ...
    </g>
    ...
    </g>
    <g id="dataarea" role="dataset">
      <g tabindex="0" transform="translate(44.444,351.22)" role="datapoint" aria-labelledby="x1">
        <rect class="bar" width="66.667" height="48.78"/>
        <text x="33.334" y="10" ...  role="datavalue" id="value1">9</text>
      </g>
      <g tabindex="0" transform="translate(155.556,243.902)" role="datapoint" aria-labelledby="x2">
        <rect class="bar" width="66.667" height="156.098"/>
        <text x="33.334" y="10" ...  role="datavalue" id="value2">20</text>
      </g>
      ...
    </g>
  </g>
</svg>
```

Listing 1: Part of an accessible SVG bar chart generated by AChart Creator. Note the semantic enrichment conveyed by applying attributes such as `aria-charttype="bar"`, `role="xaxis"`, and `role="datapoint"` to the SVG elements.

```
acreate  [--chart] CHART-TYPE
  [--dataset CSV-FILENAME] [--output SVG-FILENAME]
  [--chart-title TITLE] [--chart-desc DESCRIPTION]
  [--x-axis-title TITLE]
  [--y-axis-title TITLE]
  [--legend-title TITLE]
  [--target SOFTWARE]
  [--column DATA-COLUMN] [--no-sort]
  [--no-legend] [--no-tooltips] [--no-bar-values]
  [--no-segment-values] [--no-segment-percentages]
  [--segment-percentage-precision PLACES]
  [--svg-precision PLACES]
  [--version] [--help] [--columns]
  [--rotate-x-labels [ROTATION]]
  [--rotate-y-labels [ROTATION]] [--colors]
```

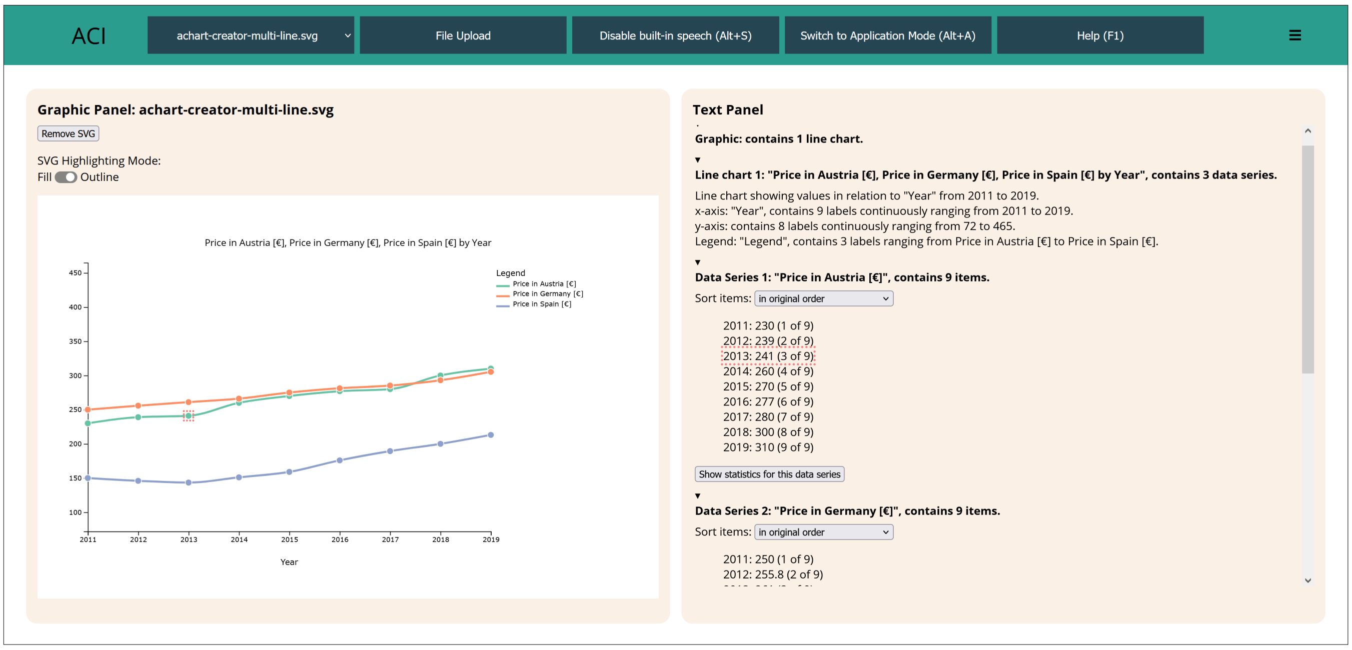
Listing 2: The syntax and options of the AChart Creator `acreate` command.

```
Fruit,Amount
Apples,9
Bananas,20
Grapefruits,30
Lemons,8
Oranges,12
```

Listing 3: The fruit dataset in CSV format.

```
acreate --chart bar --dataset fruit.csv \
  --chart-title "Most Popular Fruits" \
  --chart-desc "Each fruit with volume sold."
```

Listing 4: The AChart Creator `acreate` command to create an accessible bar chart from the fruit dataset.


ACI
achart-creator-multi-line.svg
File Upload
Disable built-in speech (Alt+S)
Switch to Application Mode (Alt+A)
Help (F1)
Price in Austria [€], Price in Germany [€], Price in Spain [€] by Year
450
400
350
300
250
200
150
100
2011
2012
2013
2014
2015
2016
2017
2018
2019
Year
Legend
Price in Austria [€]
Price in Germany [€]
Price in Spain [€]
Remove SVG
Show statistics for this data series

Graphic Panel: achart-creator-multi-line.svg

SVG Highlighting Mode:
Fill Outline

Text Panel

Graphic: contains 1 line chart.

▼
Line chart 1: "Price in Austria [€], Price in Germany [€], Price in Spain [€] by Year", contains 3 data series.

Line chart showing values in relation to "Year" from 2011 to 2019.
x-axis: "Year", contains 9 labels continuously ranging from 2011 to 2019.
y-axis: contains 8 labels continuously ranging from 72 to 465.
Legend: "Legend", contains 3 labels ranging from Price in Austria [€] to Price in Spain [€].

▼
Data Series 1: "Price in Austria [€]", contains 9 items.

Sort items: in original order

2011: 230 (1 of 9)
2012: 239 (2 of 9)
2013: 241 (3 of 9)
2014: 260 (4 of 9)
2015: 270 (5 of 9)
2016: 277 (6 of 9)
2017: 280 (7 of 9)
2018: 300 (8 of 9)
2019: 310 (9 of 9)

▼
Data Series 2: "Price in Germany [€]", contains 9 items.

Sort items: in original order

2011: 250 (1 of 9)
2012: 255.8 (2 of 9)

Figure 4: AChart Interpreter. The user has navigated to the third data point of Data Series 1.

7 Concluding Remarks

AChart defines an extended taxonomy of ARIA roles and properties, inspired by those of Describler, and uses these and standard SVG text elements to semantically enrich an SVG chart to make it accessible. AChart Creator is a command-line tool to create such accessible SVG charts from CSV files. AChart Interpreter is an application to view, explore, and read out such accessible SVG charts, which could potentially also be reworked into a browser extension. AChart currently supports bar charts, line charts, and pie charts. The project provides a working proof of concept and will hopefully contribute to the evolution of more widely accepted standards for annotating SVG charts for better accessibility.

Acknowledgements

Keith Andrews conceived of and led the AChart project. Christopher Kopel was the main developer, as part of his Master's thesis work [6], supported by Inti Gabriel Mendoza Estrada. Further contributions were made by Alexander Grass, Lea Novak, Danica Radulovic, Lukas Bodner, Daniel Geiger, and Lorenz Leitner. Later, Moritz Erlacher, Lisa Habich, Alexander Perko, and Markus Stradner worked on experimental extensions to AChart for stacked and grouped bar charts, scatterplots, and parallel coordinates.

References

[1] W3C. *Accessible Rich Internet Applications (WAI-ARIA) 1.3*. W3C First Public Working Draft. World Wide Web Consortium (W3C), Jan. 23, 2024. `https://w3.org/TR/wai-aria-1.3/`.

[2] W3C. *WAI-ARIA Graphics Module*. W3C Recommendation. World Wide Web Consortium (W3C), Oct. 2, 2018. `https://w3.org/TR/graphics-aria-1.0/`.

[3] Doug Schepers. *Describler*. 2015. `http://describler.com/`.

[4] Doug Schepers. *describler: SVG Dataviz Accessibility Tool*. Mar. 31, 2017. `https://github.com/shepazu/describler`.

[5] Doug Schepers. *Accessible SVG Data Visualization*. Feb. 18, 2015. `https://youtu.be/W1VUr544i84`.

[6] Christopher A. Kopel. "Accessible SVG Charts with AChart Creator and AChart Interpreter". Master's thesis. Graz University of Technology, Austria, May 16, 2021. `https://ftp.isds.tugraz.at/pub/theses/ckopel-2021-msc.pdf`.

[7] Nexe. *Nexe*. July 4, 2024. `https://github.com/nexe/nexe`.

[8] Michael Bostock. *D3.js Data-Driven Documents*. July 8, 2024. `http://d3js.org/`.

[9] jsdom. *jsdom*. July 14, 2024. `https://github.com/jsdom/jsdom`.

[10] Alexander Perko. *AChart Creator Showcase Video*. Graz University of Technology, Apr. 29, 2021. `https://youtu.be/NLKqTTnKLII`.

[11] Christopher A. Kopel, Keith Andrews, Inti Gabriel Mendoza Estrada, Alexander Grass, Lea Novak, and Danica Radulovic. *AChart Creator*. May 14, 2021. `https://github.com/tugraz-isds/achart-creator`.

[12] MDN. *Web Speech API*. Feb. 19, 2023. `https://developer.mozilla.org/docs/Web/API/Web_Speech_API`.

[13] Microsoft. *TypeScript: Typed JavaScript at Any Scale*. July 14, 2024. `https://typescriptlang.org/`.

[14] OpenJS. *Electron*. OpenJS Foundation. July 14, 2024. `https://electronjs.org/`.

[15] Alexander Perko. *AChart Interpreter Showcase Video*. Graz University of Technology, Apr. 29, 2021. `https://youtu.be/NLKqTTnKLII`.

[16] Christopher A. Kopel, Keith Andrews, Inti Gabriel Mendoza Estrada, Lukas Bodner, Daniel Geiger, and Lorenz Leitner. *AChart Interpreter*. May 14, 2021. `https://github.com/tugraz-isds/achart-interpreter`.

[17] Christopher A. Kopel, Keith Andrews, Inti Gabriel Mendoza Estrada, Lukas Bodner, Daniel Geiger, and Lorenz Leitner. *AChart Interpreter*. Online demo. May 15, 2021. `https://tugraz-isds.github.io/achart-interpreter/`.