Techniques for Accessible Charts

Ville Määttä, Dorian Percic, Tomo Ratko
Group G3

Information Visualisation
10.05.2023

Agenda

- Introduction
- Screen Readers
- WAI-ARIA in HTML
- WAI-ARIA Graphics Module
- SVG and Accessibility
- Experimental Tools
- Chartability
- Visa Chart Components

Introduction

11 characters

- *Accessibility* a.k.a. *a11y*
- Focus on visually impaired people.
- 18% of Americans (age > 45) are legally blind.
- 2022: 96.8% of home pages with Web Content Accessibility Guidelines (WCAG) 2.1 failures.
- Many challenges exist: Awareness, experience, etc.

https://fuzzymath.com/blog/improve-accessibility-for-visually-impaired-users/
https://webaim.org/projects/million/
https://oarep.usim.edu.my/jspui/bitstream/123456789/8393/1/Web%20Accessibility%20Challenges.pdf

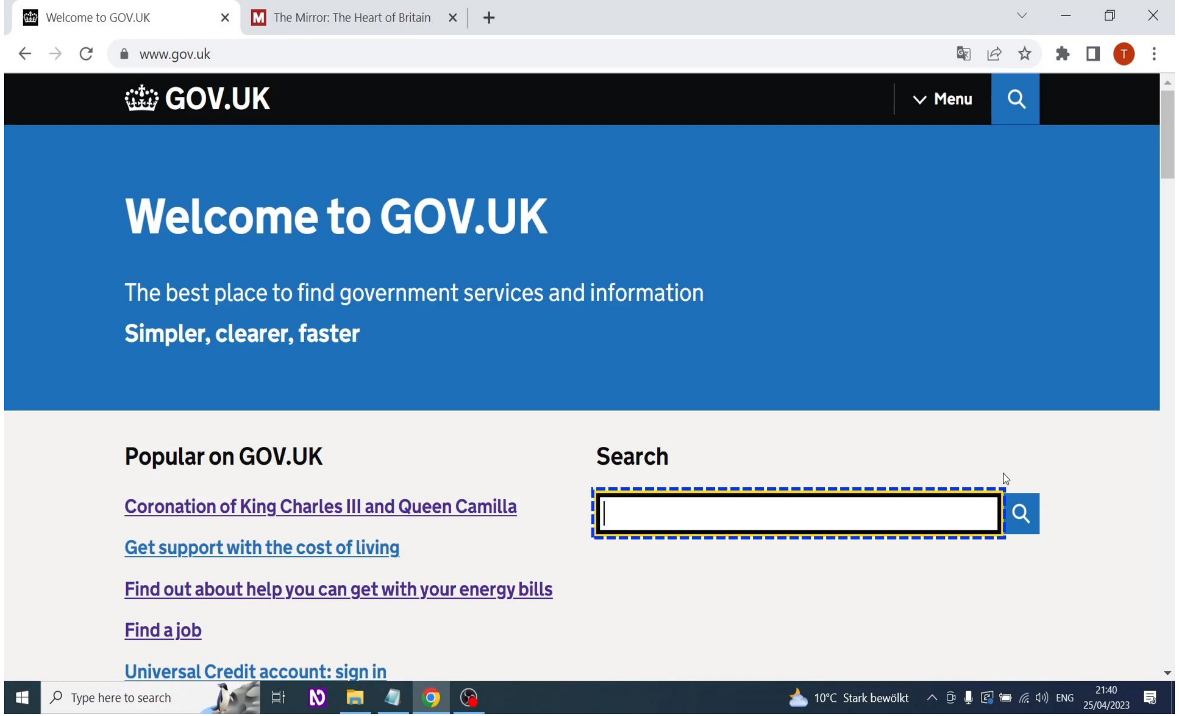
Screen Readers

Windows	JAWS (53.7%)	NVDA (30.7%)	Narrator (0.5%)
Linux	Orca	Emacspeak	Fenrir
macOS/iOS	VoiceOver (6.5%)	Voice Dream Reader	Text to Speech!
Android	TalkBack	Speechify	Voice Aloud Reader

Note: Percentages represent statistics of primary screen reader.
Data taken from: https://webaim.org/projects/screenreadersurvey8/

Live Demo

- NVDA



Welcome to GOV.UK

The best place to find government services and information

Simpler, clearer, faster

Popular on GOV.UK

- Coronation of King Charles III and Queen Camilla
- Get support with the cost of living
- Find out about help you can get with your energy bills
- Find a job
- Universal Credit account: sign in

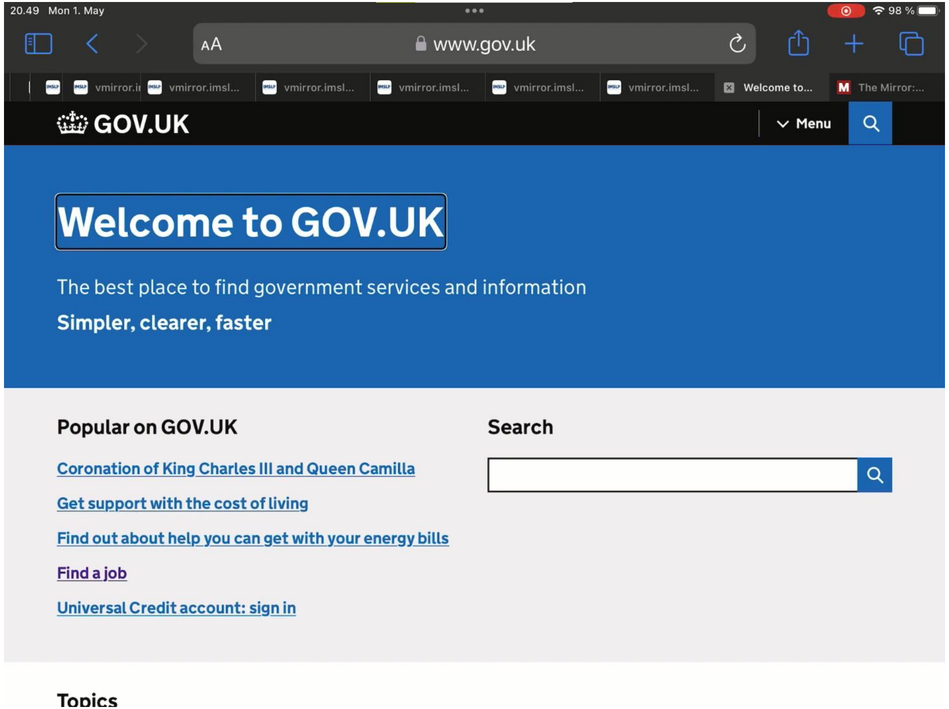
Search

Screenshot taken by Tomo Ratko on 30.04.2023.

Demo video: https://youtu.be/f5YEdqsG0DU
GitHub: https://github.com/nvaccess/nvda

VoiceOver

- Built-in screen reader for iOS.
- Different gestures possible.



Welcome to GOV.UK

The best place to find government services and information

Simpler, clearer, faster

Popular on GOV.UK

- Coronation of King Charles III and Queen Camilla
- Get support with the cost of living
- Find out about help you can get with your energy bills
- Find a job
- Universal Credit account: sign in

Search

Topics

Screenshot taken by Ville Määttä on 04.05.2023.

Demo video: https://youtu.be/oDhqWRGc--g
User guide: https://support.apple.com/guide/voiceover/welcome/mac

WAI-ARIA in HTML

- “Web Accessibility Initiative – Accessible Rich Internet Applications”
- Increase accessibility of web apps.
- Give more semantics to HTML-Elements.
- Different Roles/Properties/States are possible.
- The Accessibility Tree

https://web.dev/learn/accessibility/aria-html/
https://www.w3.org/TR/wai-aria/
https://blogs.intuit.com/2021/01/12/why-the-accessibility-tree-is-key-to-understanding-accessible-applications/

WAI-ARIA Roles

- Define function of elements.

```
<form role=”search”>
<input type=”search” id=”input” name=”text”>
</form>
```

WAI-ARIA States

- Current state of the object: Pressed, hidden, required, etc.

```
<input type=”email” id=”email”
aria-required=”true”/>
```

WAI-ARIA Properties

- Describe relationship between elements.

```
<button aria-describedby=”delete-desc”>
Delete items</button>

<p id=”delete-desc”>
Items will be permanently deleted.</p>
```

https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Roles/search_role
https://web.dev/learn/accessibility/aria-html/

WAI-ARIA Best Practices

1. Use native HTML as much as possible.

   `<button>Submit!</button>` *instead of* `<a role=”button”>Submit!</a>`

2. Interactive ARIA controls should be usable with keyboard.

   ```
   <my-custom-button tabindex=”0”>Click me!</my-custom-button>
   ```

3. Interactive elements should have accessible names (e.g. alt-text).

   ```
   <a href=”football.html”>
       <img src=”football.png” alt=”White football”>
   </a>
   ```

https://www.w3.org/TR/using-aria/#NOTES

WAI-ARIA Graphics Module

- `role=”graphics-document”`: Container for graphics objects, that can be navigated and presented as a unit.

- `role=”graphics-object”`: Part of graphics document with semantic meaning. May have nested sub-components.

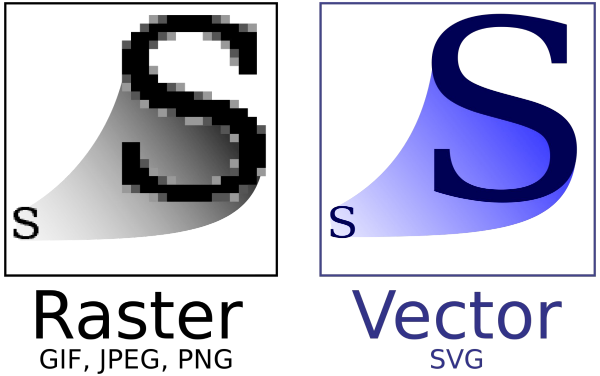
- `role=”graphics-symbol”`: Atomic graphical object to convey meaning.

```
<h2>Sport Items</h2>
<ul>
    <li> Football
        <img role=”graphics-symbol” title=”White Football” src=”football.png” />
    </li>
</ul>
```

https://www.w3.org/TR/graphics-aria-1.0/

Scalable Vector Graphics (SVG)

- A means to describe vector graphics in markup language on a web page.
- Graphic format based on vectors instead of pixels:
  - Scalability without loss of information.
  - Inefficient for more complex graphics.



https://www.w3.org/TR/2011/REC-SVG11-20110816/

Image taken from: https://en.wikipedia.org/wiki/SVG#/media/File:Bitmap_VS_SVG.svg and is under *Attribution-ShareAlike 2.5 Generic (CC BY-SA 2.5) license.*

SVG and Accessibility

- WAI-ARIA Graphics Module tags can be applied to SVG elements.
- SVG charts can be made accessible:
  - By adding descriptive text (manually or with AI) and place it in `<desc>` element.
  - By adding ARIA roles and properties.

```
<svg version=”1.1” viewBox=”0 0 500 500” aria-labelledby=”title”
aria-describedby=”desc”>
     <title id=”title”>Accessible bar chart</title>
     <desc id=”desc”>Short description of the chart</desc>
     <g role=”table”>
     …
     </g>
</svg>
```

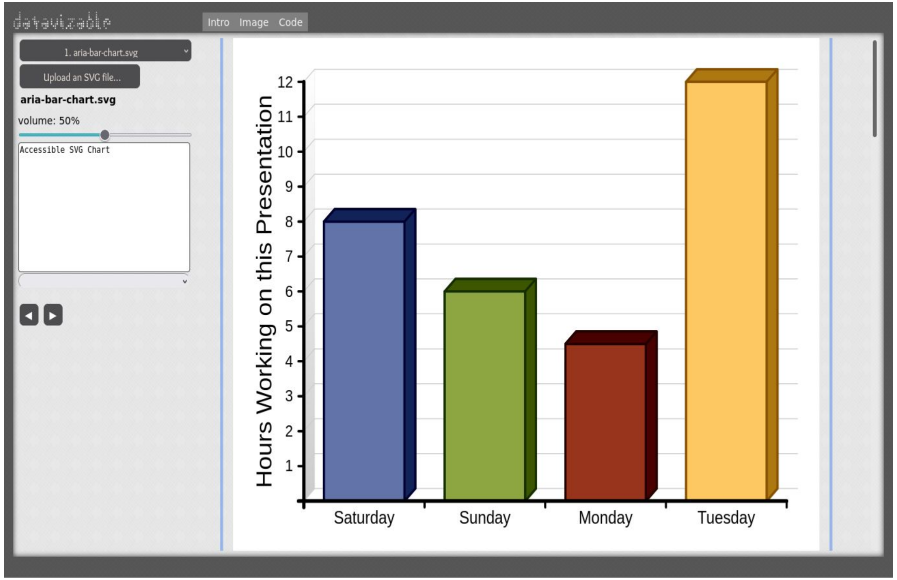
https://www.urban.org/sites/default/files/2022-12/Do%20No%20Harm%20Guide%20Centering%20Accessibility%20in%20Data%20Visualization.pdf

Experimental Tools

- Describler
- AChart Creator/Interpreter

Describler

- Prototype SVG chart reader.
- Advantages:
  - Web application: Cross platform
  - Intuitive interaction
  - Default examples provided
- Disadvantages:
  - Many bugs (e.g. importing)
  - No documentation on custom aria roles/properties.

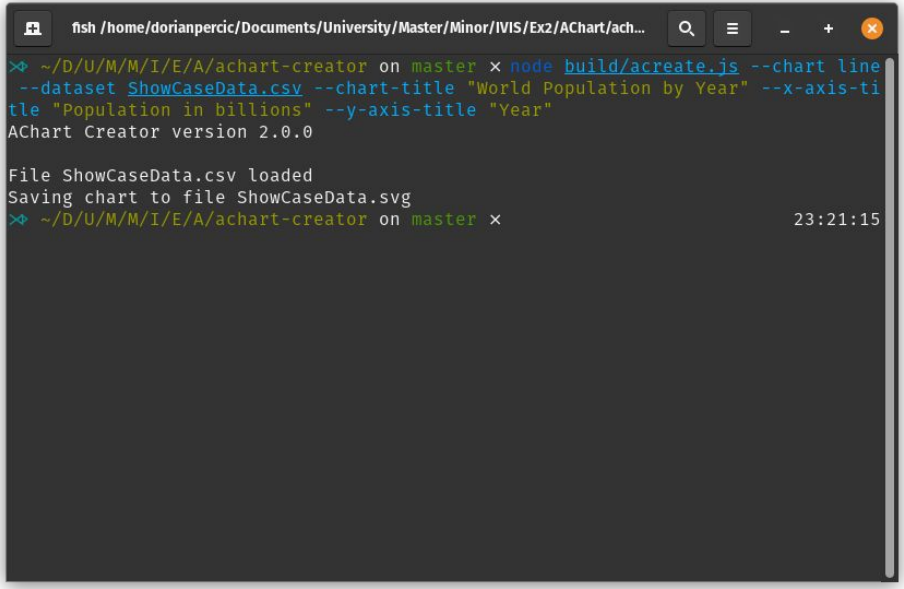


Screenshot taken by Dorian Percic on 01.05.2023.

Demo video: https://youtu.be/WR8n80OvT_0
Website: describler.com

AChart Creator

- Command-line tool
- Create accessible SVG charts
- Advantages:
  - Many chart options
  - Good documentation
  - Cross platform
- Disadvantages:
  - No re-editing of existing charts.
  - For non-technical people difficult to use.


fish /home/dorianpercic/Documents/University/Master/Minor/IVIS/Ex2/AChart/ach...

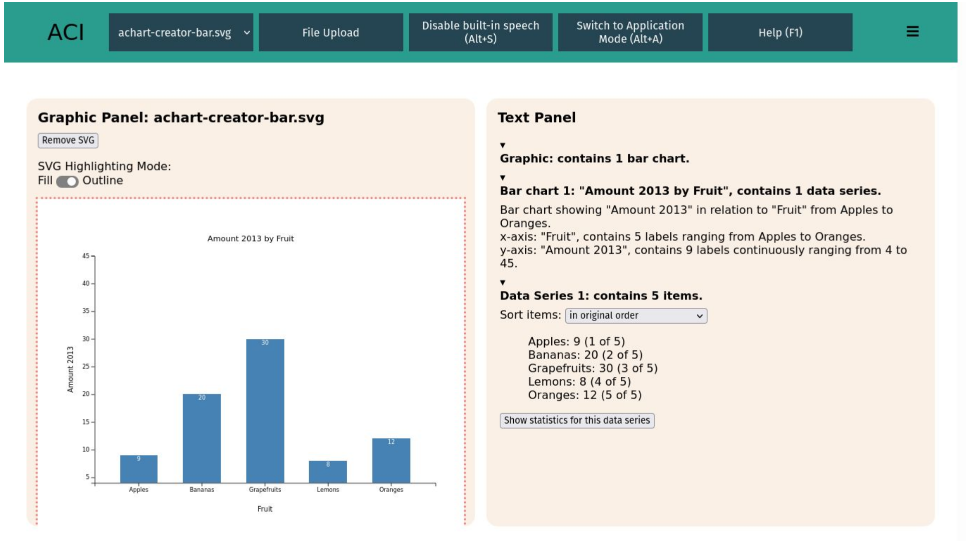
```
⋊> ~/D/U/M/M/I/E/A/achart-creator on master × node build/acreate.js --chart line
 --dataset ShowCaseData.csv --chart-title "World Population by Year" --x-axis-ti
tle "Population in billions" --y-axis-title "Year"
AChart Creator version 2.0.0

File ShowCaseData.csv loaded
Saving chart to file ShowCaseData.svg
⋊> ~/D/U/M/M/I/E/A/achart-creator on master ×                           23:21:15
```

Screenshot taken by Dorian Percic on 30.04.2023.

Demo video: https://youtu.be/CsHPDiHHCCc
GitHub: https://github.com/tugraz-isds/achart-creator

AChart Interpreter

- Charts in SVG
- Textual Summary
- Advantages:
  - Cross platform
  - General statistics
  - Good documentation.
- Disadvantages:
  - Image in graphics panel cut without full-screen mode.



Graphic: contains 1 bar chart.

Bar chart 1: "Amount 2013 by Fruit", contains 1 data series.

Bar chart showing "Amount 2013" in relation to "Fruit" from Apples to Oranges.
x-axis: "Fruit", contains 5 labels ranging from Apples to Oranges.
y-axis: "Amount 2013", contains 9 labels continuously ranging from 4 to 45.

Data Series 1: contains 5 items.

Sort items: in original order

Apples: 9 (1 of 5)
Bananas: 20 (2 of 5)
Grapefruits: 30 (3 of 5)
Lemons: 8 (4 of 5)
Oranges: 12 (5 of 5)

Screenshot taken by Dorian Percic on 30.04.2023.

Demo video: https://youtu.be/QlX3VdTL1t0
GitHub: https://github.com/tugraz-isds/achart-interpreter

Live Demo

- AChart Creator

- AChart Interpreter

Chartability

- A set of heuristics for ensuring that data visualizations, systems, and interfaces are accessible.
- Based on WCAG 2.1.
- 7 principles:
  - Perceivable, Operable, Understandable and Robust (from WCAG)
  - Compromising, Assistive and Flexible
- Accessibility as a scale rather than a state.

GitHub: https://github.com/Chartability/POUR-CAF

Website: https://chartability.fizz.studio/

Visa Chart Components

- Built using D3.
- Common charts (e.g. bar chart, line chart, etc.) with accessibility features.
- Meets Visa’s Global Accessibility Requirements (VGAR) standards.

GitHub:https://github.com/visa/visa-chart-components

Website: https://visa.github.io/visa-chart-components/?path=/story/introduction--page

DataViz Accessibility Advocacy and Advisory Group

- https://github.com/dataviza11y/Why-We-Exist

Thank you for your attention! Any questions?