WCAG 2.0 for Designers: Beyond Screen Readers and Captions

Loretta Guarino Reid[1] and Andi Snow-Weaver[2],

[1]Google, Inc
1600 Amphitheatre Pkwy
Mountain View, CA 94043
LorettaGuarino@google.com

[2]IBM
11501 Burnet Rd.
Austin, TX 78726
andisnow@us.ibm.com

Abstract. The W3C Web Content Accessibility Guidelines (WCAG) provide guidance on making websites accessible to people with disabilities. WCAG 1.0 focused largely on coding requirements that enable websites to interoperate with assistive technologies used by people with disabilities. WCAG 2.0 addresses an environment where website complexity has increased significantly due to higher network bandwidth and the introduction of new interactive technologies. It places more constraints on the default look and feel of a website. Of the 38 Level A and AA provisions, about 50%, impact the website design. This paper reviews those requirements, examining the user needs that they are intended to support and highlighting example strategies for addressing those needs.

Introduction

The W3C Web Content Accessibility Guidelines (WCAG) provide guidance on making websites accessible to people with disabilities. WCAG 1.0 [1], the first version of the guidelines, became a W3C Recommendation in 1999 and was very specific to HTML, the dominant Web technology at that time. Its successor, WCAG 2.0 [2], became a W3C recommendation in December 2008. WCAG 2.0 was designed to be a set of testable technology neutral requirements applying to the wide and continuously evolving range of Web technologies now available. Thus the requirements in WCAG 2.0 look very different than WCAG 1.0.

WCAG 1.0 requirements describe the strategy to be used to address a particular problem. In contrast, WCAG 2.0 requirements are general statements of desired outcomes to be achieved but allow flexibility in the strategies used to address them. For example, WCAG 1.0 contains very specific requirements for data tables, lists, headings, etc. The reason for these requirements is that the visual presentation of the elements conveys meaning that must be preserved when the page is rendered, for instance by an assistive technology in an audio modality instead of a visual one.

But there are other types of content that WCAG 1.0 does not address where the visual presentation conveys meaning. To close this gap and cover all such cases, WCAG 2.0 generalized the requirement to the desired outcome: *Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text*. The WCAG 1.0 requirements are important strategies for achieving this outcome but they are not the only ones. The WCAG 2.0 requirement is applicable to anything where the visual presentation has meaning.

Since all Web developers cannot be expected to be experts in the strategies available to achieve a required outcome, the W3C has provided the following WCAG 2.0 companion documents to further explain each requirement and provide suggested strategies for achieving it:

- Understanding WCAG 2.0 [3]
- How to Meet WCAG 2.0 [4]

Another significant difference between WCAG 1.0 and WCAG 2.0 is the degree of focus on assistive technology support, which is largely a coding consideration, vs. designing for accessibility.

Supporting Assistive Technologies

WCAG 1.0 focuses largely on coding requirements that enable websites to interoperate with assistive technologies used by people with disabilities. At the time WCAG 1.0 became a W3C Recommendation, the majority of websites were simple information sites. Most consisted of some navigation links across the top or down the left side with a topical article being the main focus of the page. Users interacted with websites by selecting links or perhaps entering some information into a form and submitting it with a button. Interaction required sending a request to a server and waiting for the next page to load into the browser.

Since the design of websites was simple, network bandwidth was low and there wasn't much interactivity, coding for interoperability with assistive technology was generally sufficient for compliance with WCAG 1.0. Of the 46 priority 1 and 2 requirements, only about 35% impact the design of the look and feel of a site. As such, designers were not so involved in website accessibility, leaving its implementation to programmers and verification to testing tools. Websites were overwhelmingly implemented in HTML, so this was generally a valid approach. Ten years ago, it was usually possible to make static HTML websites WCAG 1.0 AA conforming without much impact to the design. Where design changes needed to be made, they were usually minor.

Beyond Assistive Technology Support

Since then, however, website complexity has increased significantly due to higher network bandwidth and the introduction of new interactive technologies. Interoperability with assistive technology remains critically important. Therefore, WCAG 2.0 also contains provisions for providing the necessary information to assistive technologies to present content in a variety of modalities. Even though a complex site

might be coded to be interoperable with assistive technologies, it could still be difficult for users with disabilities to use unless their needs were considered in the initial design of the site.

Taking this issue into consideration, WCAG 2.0 places more constraints on the default look and feel of a website. Of the 38 Level A and AA provisions, about 50% impact the website design, most to a greater degree than the design impacts of WCAG 1.0. Consequently, in order to conform to WCAG 2.0, website designers must consider accessibility when designing a site.

Like its predecessor, WCAG 2.0 defines three groups of requirements, Level A, AA, and AAA with Level A being the minimum level of conformance. Level A and AA requirements are those that are applicable to all websites. The testable requirements in WCAG 2.0 are called *success criteria*. References to specific WCAG 2.0 success criteria in this paper will be of the form SC X.X.X. This paper explores the WCAG 2.0 Level A and AA requirements that potentially impact the visual, auditory, and interaction design of a website, examining the user need that they are intended to support and highlighting example strategies that might be used. Level AAA requirements that affect design are not covered in this paper.

Many of these requirements are simplified in this overview, and readers are advised to consult WCAG 2.0 [2] and Understanding WCAG 2.0 [3] for the details of the requirements. Those who are familiar with WCAG 1.0 will also find the *Comparison of WCAG 1.0 Checkpoints to WCAG 2.0* [5] useful.

Visual and Auditory Design

Visual design encompasses many aspects of a website including the color scheme, size of text, layout of the components, and the use of color or movement to attract the user's attention. With the rise of high speed networks, more and more sites are also including audio to enhance the user experience of their site. All of these things can impact a user with a disability.

For users with vision impairments who do not use assistive technology, color, contrast, and text size are critically important. They may find instructions that rely on the user's ability to see the spatial relationships on the display impossible to follow. And they cannot easily scan a page to search for visual cues such as icons used to tag fields in error.

Blind users depend solely on the audio version of the page as rendered by their screen reader software, so audio played automatically when a page is loaded can interfere with their ability to perceive the information on the page.

Users with mobility impairments who use only the keyboard to operate a site must be able to see the location of the keyboard focus. And completing forms can be an especially difficult task for users with disabilities, so care must be taken to consider their needs when designing forms.

We shall review some of the WCAG 2.0 requirements that address these needs.

Color (SC 1.4.1)

While WCAG 1.0 allows information to be conveyed through color as long as the color is available through markup, WCAG 2.0 requires that color not be the only *vis-*

ual means of conveying information. Strategies for meeting this requirement include providing text or text cues, in addition to the color cues, or using different patterns and textures in addition to different colors.

Contrast (SC 1.4.3)

WCAG 1.0 requires that there be "sufficient contrast" for text and background color combinations which is a subjective requirement. WCAG 2.0 specifies a particular minimum contrast ratio of 4.5 to 1. This testable benchmark provides a level of contrast to people with mild visual impairments that is comparable to the 3:1 minimum level recommended for unimpaired vision [6] [7]. A variety of tools are available for measuring the contrast between various color combinations.

Text size (SC 1.4.4)

WCAG 2.0 requires that text be resizable up to 200%, so that people with mild vision impairments can read the text without need of a screen magnifier. WCAG 1.0 required the use of relative rather than absolute units, which is often a useful technique for supporting resizable text. However, depending upon the technology used and the capabilities of the user agent, there can be other ways to support larger text. Many current browsers provide a zoom function that enlarges the entire page, not just the text.

Instructions and sensory capabilities (SC 1.3.3)

Instructions must not depend on the user's ability to see the content as it is presented on a display device, such as "Press the button on the right." The spatial information may be helpful and should be included. However, additional information must also be included for users who can't see which button is "on the right". For example, "Press the 'submit' button on the right."

Control of audio that plays automatically (SC 1.4.2)

Web pages that automatically play audio when accessed can interfere with the screen reader audio relied upon by blind users. Any audio that plays automatically must stop playing in less than three seconds or there must be a mechanism for the user to stop it. This requirement ensures that the interference stops quickly or that the user can stop it so he or she can listen to the rest of the content on the page with their screen reader.

Visible keyboard focus indicator (SC 2.4.7)

Users who rely on the keyboard to operate a web page need a clear visual keyboard focus indicator so they can determine which component will react to keyboard commands. WCAG 2.0 contains a new requirement to address this issue. Design your web page to take advantage of built-in support in browsers, when possible, or provide a custom focus highlight as part of your web design.

Triggering seizures (SC 2.3.1)

Certain types of moving content can trigger seizures in people with photosensitive epilepsy. While WCAG 1.0 required that the screen not flicker to avoid triggering seizures, WCAG 2,0 spells out more clearly what types of flashing content must be avoided. Content that does not flash more than three times in one second will not trigger seizures. Flashing content that occupies a very small area of the display screen also does not cause seizures. Even flashing content that occupies a large area of the display screen does not cause seizures unless it involves certain color combinations. WCAG 2.0 provides measurable criteria for determining whether or not flashing content is acceptable. In contexts where flashing content is desirable, tools are available to evaluate it for conformance with WCAG 2.0.

Labels or instructions where user input is required (SC 3.3.2)

Labels and instructions help users understand how to complete forms and may be critical for screen reader and magnifier users and those with cognitive disabilities. It is not necessary to provide a visible label for every form field, however, either labels or instructions must be provided. Position labels near the fields they label so that screen magnifier users will be able to see them near the field itself. Examples of expected data formats (for example, *mm/dd/yy* for a date field) qualify as instructions.

Consistent identification for recurring function (SC 3.2.4)

Users with disabilities take more time to learn to use a site they have never visited before but can become quite efficient once they are familiar with the site. For example, they will use search strategies to quickly locate a function that is expected to occur frequently throughout a site. Label recurring function consistently to help users become more efficient as they use your site.

Interaction

Interaction design is also very important for users with disabilities; both the design of how one operates the site and how one comes to understand how to operate the site. The most basic aspect of interaction is the user's input device. Designers usually assume that users can use a mouse but many users with disabilities can only use a keyboard or specialized input device that mimics a keyboard. With keyboard operation, the order in which objects receive focus is important. In addition, the context should not change in unpredictable ways. In WCAG 1.0, it was assumed that the user agent or browser was responsible for keyboard operation. But with many of the Web technologies now available, website designers must address interaction issues in their designs.

Other interaction design decisions that impact users with disabilities are time limits for completing tasks and the design of navigation mechanisms. Users with disabilities may need a larger amount of time to complete a task and may not detect when information on the page has been updated. They need navigation mechanisms presented in a consistent manner and they need help avoiding or correcting errors.

We shall review some of the WCAG 2.0 requirements in these areas.

Keyboard operation (SC 2.1.1 and SC 2.1.2)

All of the functionality of the site or application must be operable using only the keyboard. Many times this requirement can just be handled in the coding of the web site but there are scenarios where it impacts the website design. For example, server side image maps only support interaction via a mouse. So an alternative method that is keyboard accessible must be included in the design. On pages that mix technologies, some of which are not accessible, the keyboard focus can become trapped in inaccessible content. Provide a means to avoid such inaccessible content or instructions for escaping from it.

Logical sequential focus order (SC 2.4.3)

Where websites can be navigated in sequential order, e.g. via the Tab key, and that order affects the meaning or operation of the page, a focus order that is consistent with the meaning is essential. If the meaning is affected by the focus order, the designer should specify it so that the programmers can implement it properly. For example, if a form consists of several sections, the focus order should move from section to section, rather than skipping back and forth between sections In general, users will expect focus to follow the natural reading order.

Unpredictable changes of context (SC 3.2.1 and SC 3.2.2)

WCAG 1.0 contained a provision that restricted the use of spawned windows. Spawned windows can be confusing to users with disabilities because they change the context. But spawned windows are not a problem as long as they don't occur unexpectedly. Furthermore, spawned windows are only one example of a change of context. So rather than address the specific case of spawned windows, WCAG 2.0 addresses changes of context generically and only where they are unexpected such as when the keyboard focus is moved into an interactive component or when the user makes a selection in a component. In contrast, changes of context that occur when selecting a link or button are expected and therefore are not prohibited. Designers may have assumed that the user would explore a form or application visually and only interacts with a component when he or she is certain of their choice. But screen reader users and keyboard only users explore a form or application by navigating to all of the interactive components sequentially. If moving keyboard focus to a component or selecting a value from a set of options causes a form to be submitted or a new window to be opened, it can be disorienting or result in the form being submitted unintentionally. In certain technologies, this can make it impossible for keyboard users to complete forms. Design your site so that changes of context occur only when users take actions such as selecting a button or a link.

Time limits (SC 2.2.1)

Users with disabilities often require significantly more time to complete a task than users without disabilities. They may be slowed down by their use of assistive technology. Screen reader users can't scan the form quickly and complete only the required fields or those of interest. They have to read sequentially through all of the fields in a form. Some mobility-impaired users only have the ability to press a single button or switch. They use software that scans through each key on an onscreen keyboard until

the user activates the button or switch to select the desired character or function key. And some users with reading disabilities need a lot more time to read and understand information and instructions. It is best if time limits can be avoided altogether but this may not be possible due to limited resources or security exposures. If they can't be avoided, provide settings that allow users to disable time limits or adjust them. Or warn the user that a time limit is about to expire and allow him or her to extend it in order to complete the task. For WCAG 2.0 AA conformance, exceptions are allowed for time limits that are essential to the task, for real-time events such as an auction, or time limits of more than 20 hours.

Moving or auto-updating information (SC 2.2.2)

Content that is moving or auto-updating can be a problem for anyone who has trouble reading text quickly or who is easily distracted by motion. It also causes difficulties for screen reader users. Better design is to provide a way for users to start such activity explicitly, rather than starting automatically. When it does start automatically, there must be a way for users to stop or pause the activity.

Consistent order of navigation (SC 3.2.3)

Just as they benefit from consistent identification for recurring functions, users with disabilities benefit from a familiar order to navigation elements that are repeated on different pages within the site. Consistency makes it easier to navigate within and interact with the page, particularly for users who cannot scan the entire page quickly and easily. Navigation bars, menus, and other sets of interactive elements should occur in a consistent order throughout the website or application.

Errors (SC 3.2.2, SC 3.3.1 and SC 3.3.3)

Detecting and correcting errors is a particularly problematic task for users with disabilities. When visual cues are used to indicate fields that contain input errors, the assumption is that the user can scan the form quickly to locate the errors. But for users who are unable to scan the form quickly, this is a tedious task of re-navigating the form to search for the error indications. These users may become so frustrated that they abandon the task altogether. Of course visual cues are helpful to users who can see and should be used where appropriate, however, also provide a text message that describes the error as specifically as possible along with suggestions for correcting the error. Errors in legal or financial transactions or in tasks that result in the deletion of data can have serious implications such as purchasing the wrong product or submitting an unintentional bid in an auction. WCAG 2.0 requires one of three strategies for this important scenario: 1) the transaction must be reversible, 2) the information entered by the user must be checked for errors and the user must be provided an opportunity to correct the errors, or 3) the user must be given an opportunity to review the information entered and change it before committing the transaction.

Accessibility Support for Uses of a Web Technology

In addition to meeting the individual provisions that impact the design, designers who specify the implementation technologies must also understand the WCAG 2.0 concept of accessibility support.

WCAG 2.0 does not require the use of particular technologies. Neither does it prohibit the use of particular technologies by requiring that a site be usable when they are disabled. WCAG 1.0, in contrast, requires that documents can be read without stylesheets, and that pages be usable when scripts or applets are turned off or not supported. Nor are there any WCAG 2.0 provisions that require workarounds "until user agents" support a particular function as there are in WCAG 1.0.

However, designers must not assume that every Web technology can be accessed by people with disabilities using assistive technologies. Rather they must investigate the current level of accessibility support of each technology they intend to use. For WCAG 2.0 conforming content, they may only use technologies in ways that are accessibility supported by the browsers and assistive technologies available to their users. Where new technologies that are not accessibility supported are required or desired, they may still be used as long as alternative versions of the information and functionality are provided in ways that are accessibility supported.

WCAG 2.0 introduced the concept of accessibility support because it recognized that new technologies continue to emerge on the web. It is important to provide a pathway for new technologies to become available for users with disabilities. Even when these technologies provide the features needed for accessibility, it may take time for assistive technologies to catch up. Authors may provide content using a new technology for users who have the necessary support as long as they do not rely upon the technology until it is accessibility supported. In other words, when using newer technologies that are not accessibility supported, be sure to design equivalent versions of the content and function using technologies that are accessibility supported.

Authors are encouraged to rely upon the fewest technologies possible and to use progressive enhancement techniques to incorporate newer technologies that have inconsistent accessibility support.

> **Comment [als1]:** ???? Does the number really matter as long as they're accessibility supported?

Conforming Alternate Versions

Designers will find WCAG 2.0 more flexible in allowing multiple strategies to be used to achieve a desired functional outcome. WCAG 1.0 permitted a link to an alternate page that was accessible and used W3C technologies. WCAG 2.0 permits alternate versions of web pages as long as the user with disabilities can find the conforming alternate version. Some alternate versions may even be specialized for particular disabilities, such as versions written to different reading levels or that use a visual layout that is easier for users with certain cognitive disabilities.

There are a variety of mechanisms that can be provided to find a conforming version. A nonconforming page could contain a link to the conforming page, as long as the link itself conforms to WCAG 2.0. Or a nonconforming page could contain controls to change to a style that conforms provided the controls used to change the style conform to WCAG 2.0. This technique can be particularly useful for achieving the re-

quired contrast ratio on sites where colors that are required for certain visual designs do not meet the contrast ratio. Or there may be user settings that apply to the entire web site that select the version to be displayed.

Conclusions

In addition to requirements that support access to a web site via assistive technology, WCAG 2.0 contains a variety of requirements that affect the visual, auditory, and interaction design of a web site. These requirements must be applied during the design of the site so that users with disabilities will have effective access.

WCAG 2.0 defines these design requirements more clearly than WCAG 1.0, but provides more flexibility in how to implement them. The companion documents, Understanding WCAG 2.0 and How To Meet WCAG 2.0, provide suggested strategies and techniques for addressing these requirements. These are evolving documents that will grow as designers develop additional, innovative ways to satisfy WCAG 2.0 requirements, possibly with new web technologies. The result will be an innovative, evolving web that is accessible to all.

References

1. Chisolm, W., Vanderheiden, G., Jacobs, I.: Web Content Accessibility Guidelines 1.0. W3C Recommendation (1999) http://www.w3.org/TR/WAI-WEBCONTENT/
2. Caldwell, B., Cooper, M., Guarino Reid, L., Vanderheiden, G.: Web Content Accessibility Guidelines 2.0. W3C Recommendation (2008)
3. Caldwell, B., Cooper, M., Guarino Reid, L., Vanderheiden, G.: Understanding WCAG 2.0: A guide to understanding and implementing WCAG 2.0. W3C Note (2008)
4. How to Meet WCAG 2.0: A customizable quick reference to Web Content Accessibility Guidelines 2.0 requirements (success criteria) and techniques. W3C Note (2008)
5. Comparison of WCAG 1.0 Checkpoints to WCAG 2.0, in Numerical Order. http://www.w3.org/WAI/WCAG20/from10/comparison/
6. ANSI/HFES 100-2007, Human Factors Engineering of Computer Workstations (2007)
7. ISO 9241-304:2008, Ergonomics of human-system interaction -- Part 304: User performance test methods for electronic visual displays (2008)