# WCAG 2.0:
# A Web Accessibility Standard for the Evolving Web

Loretta Guarino Reid
Google, Inc.
1600 Amphitheatre Pkwy
Mountain View, CA 94043
+1-650-253-4380
LorettaGuarino@google.com

Andi Snow-Weaver
IBM
11501 Burnet Rd.
Austin, TX 78726
+1-512-838-9903
andisnow@us.ibm.com

## ABSTRACT

Since the Web Content Accessibility Guidelines 1.0 (WCAG) became a W3C recommendation in May 1999, the Web has changed dramatically. This paper describes some of the major issues encountered because of these changes and the approaches developed to address them in WCAG 2.0.

## Categories and Subject Descriptors
K.4.2 [**Social Issues**]: Assistive technologies for persons with disabilities – *accessibility, standards.*

## General Terms
Standardization, Legal Aspects.

## Keywords
Accessibility, WCAG.

## 1. INTRODUCTION

In the late 1990s, the World Wide Web Consortium's (W3C) Web Accessibility Initiative (WAI) set out to provide an international standard defining Web content accessibility for people with disabilities. The result was three standards: one for content [1], one for user agents (browsers and assistive technologies) [2], and one for authoring tools [3].

The first version of the guidelines, WCAG 1.0, was finalized in May of 1999 when the Web consisted mostly of static information websites and a few e-commerce sites implemented using HTML. Some more interactive technologies were starting to emerge but they were not well-supported by either browsers or assistive technologies. The strategy for WCAG 1.0 was to describe how to make HTML websites accessible to people with disabilities and require alternatives for most other types of content. Some of the accessibility features of HTML were also not well-supported. To deal with these types of issues that were expected to improve over time, WCAG 1.0 contains a number of provisions that begin with the phrase "Until user agents …"

As WCAG 1.0 began to be enforced by government policies, however, some issues arose. New technologies or previously unsupported features of HTML could not be relied upon even when well-supported by browsers and assistive technologies because the standard still required HTML alternatives and workarounds. And some of the requirements in WCAG 1.0 are so subjective that one cannot make a legally defensible claim of conformance to them. In fact, the issue of testability was one of the major reasons the US government did not adopt WCAG 1.0 as the Web accessibility standard for Section 508 [4].

Recognizing that WCAG 1.0 would become outdated, the W3C formed a working group in 2000 to develop WCAG 2.0 [5] as the second version of the W3C Web Content Accessibility Guidelines.

In the last eight years, the Web has changed dramatically. It's no longer an HTML-only world. It has evolved into an exciting, compelling medium for providing innovative services. One of the major goals of WCAG 2.0 was to describe the requirements for Web content accessibility in technology neutral language so that it could be applicable to any W3C or non-W3C technology, such as CSS, SMIL, SVG, XML, PDF, or Flash in addition to HTML and XHTML. A second major goal of WCAG 2.0 was to ensure that the requirements are all objectively testable so that policy makers can adopt them unchanged.

This paper describes some of the major issues encountered by the WCAG working group and the strategies developed to address them.

## 2. PRINCIPLES, GUIDELINES AND SUCCESS CRITERIA

So how does WCAG 2.0 go about defining the requirements for Web accessibility in such a way that they are applicable to any Web technology, are objectively testable, and can address the rapidly changing landscape of the Web?

The first step is to get the technology-specific language out of the standard itself. WCAG 2.0 requirements are organized around four general principles of accessibility:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
W4A2008 - Technical, April 21-22, 2008, Beijing, China. Co-Located with the 17th International World Wide Web Conference.
Copyright 2008 ACM ...$5.00.

- Information and functionality must be presentable to users in ways that they can perceive.
- Interactive functionality must be available to users in ways that they can operate.
- Information and functionality must be understandable.
- Information and functionality must be able to be rendered reliably by a wide variety of user agents, including assistive technologies.

A number of guidelines—which define goals and provide a framework to help authors understand the requirements—support each principle. There are a total of 12 guidelines in the current draft.

For each guideline, a number of testable requirements, called success criteria, are defined. These success criteria are worded as general conditions that need to be satisfied. In this case, the word "testable" does not imply "machine testable." Some requirements are suitable for verification using automated test tools and many require human testing. But the criteria are worded in such a way that a human can verify whether or not a website passes or fails each criterion.

In order to leave room for innovative solutions, WCAG 2.0 does not prescribe exactly *how* the conditions are to be met in the normative standard. It does, however, provide informative guidance on currently known methods for meeting the success criteria. These general and technology-specific techniques are provided in separate informational documents [6] [7]. Keeping the techniques separate makes the normative standard concise and allows for future enhancements of techniques as required without the need to change the standard. (Changing the standard involves the lengthy W3C standards approval process.) Authors who are familiar with WCAG 1.0 will recognize many of the HTML-specific techniques for WCAG 2.0.

Like WCAG 1.0, WCAG 2.0 requirements (success criteria) are organized into three groups. However, the groups have slightly different meanings. In WCAG 1.0, the groups are called *priorities* and define what were thought at that time to be the importance of the requirement in removing accessibility barriers. Experience has taught us, however, that even the priority 3 requirements are critical for some types of disabilities. WCAG 2.0 therefore uses a different scheme. The groups are simply called levels – A, AA, and AAA. The Success Criteria were assigned to one of the three levels of conformance by the working group after taking into consideration a wide range of interacting issues. Some of the common factors evaluated when setting the level included:

- possibility that assistive technology can make the content accessible
- applicability to all websites and types of content
- skills that are reasonable for authors to achieve
- limits on function, presentation, freedom of expression, design or aesthetics
- the existence of workarounds

In WCAG 2.0, the levels are a sort of *implementation order* for authors; that is, Level A is always required and Level AA provisions must also be met in order to achieve Level AAA conformance. But they do not imply priority with regard to the needs of users with disabilities.

## 3. CONTENT IS NO LONGER JUST CONTENT

In the WCAG 1.0 timeframe, there was a fairly clear distinction between content, user agents, and authoring tools and WAI provided an accessibility standard for each. But scripting technologies, now widely used on the Web, give content authors the responsibility for user agent functionality. And with the rapid rise in blogs, wikis, and social networking sites, users have become authors and designers. The lines between content, user agents, and authoring tools have blurred. It was challenging to determine how to handle this new environment in WCAG 2.0.

For requirements where the responsibility for a particular success criterion might lie with either the user agent or the content author, language such as "a mechanism is available" is used. If the mechanism is provided by user agents, then the content author can rely on that and simply ensure that the content is authored in a way that works with the user agent mechanism. But if the mechanism is not provided by the user agent for the particular technology being used, the content author has to provide it in order to comply.

In cases where information needs to be exposed for interoperability with a user agent or assistive technology, language such as "programmatically determined" is used. For these requirements, the author needs to provide enough information to make user agent and alternative rendering with an assistive technology possible.

If a Web page is an application where users can actually create content, it should also satisfy the W3C WAI Authoring Tools Accessibility Guidelines. [3]

## 4. CONTENT COMES FROM EVERYWHERE

When a user visits a page that conforms to WCAG 2.0, he or she should know that the entire page will have the properties described by WCAG 2.0: non-text content will have text alternatives; the content will be keyboard accessible; synchronized media will have captions; etc. The webmaster of the website that contains the page is the obvious candidate to make a WCAG 2.0 conformance claim for a particular page, since the webmaster posted the page to the Web.

However, in the Web 2.0 world, the webmaster may not have provided all the content on the Web page. Users add comments to blogs. Retail sites include product reviews submitted by users. News portals include feeds of content from a variety of sources. A Web-based e-mail application displays e-mail received from a variety of users.

The webmaster may have ensured that his or her parts of the page conform to WCAG 2.0. However, the content provided from other sources may contain a variety of accessibility barriers. This would be a problem for the user who expected to use the Web page based on its WCAG 2.0 conformance.

What leverage does the webmaster have with the suppliers of such content? It may be possible to provide the mechanisms for creating or adding content in a way that ensures or encourages WCAG 2.0 conformance. At the very least, those mechanisms should not prevent the supplier from providing accessible content. For instance, a site that permits users to upload pictures or videos should have some way for them to provide text alternatives or captions. To encourage conformance, the site might even prompt the user for accessibility information such as text alternatives for uploaded images or transcripts of podcasts.

Webmasters may also monitor the content for accessibility problems, and either post-process the content to fix those problems or remove content that does not conform. This is similar to the types of monitoring done for spam or offensive content.

If the webmaster has contracted for content that is being included, the contract should include requirements to meet the WCAG 2.0 standard.

However, in many circumstances, it will not be possible for the webmaster to provide a guarantee that content from other sources will conform to WCAG 2.0. The user of a Web e-mail site may not want e-mail blocked or edited because it doesn't conform.

Pages that include such content cannot claim to conform to WCAG 2.0. However, webmasters may issue a Statement of Partial Conformance. The parts of the Web page that come from some other source and that may not conform are described in a way that the user can identify them, e.g., the e-mail messages in a Web-based e-mail application, or the comments in a blog. The Statement of Partial Conformance asserts that if those portions of the page are excluded, then the Web page conforms to WCAG 2.0.

## 5. BEYOND HTML

One question that is often raised is what technologies are accessible? For instance, is it a requirement that the content be in HTML? Must the Web page work if CSS and Javascript are not enabled? Can Flash content be accessible? Since SVG provides better accessibility for images, must SVG be used for images? Must math be represented in MathML?

Because WCAG 2.0 is technology neutral, it does not answer this question directly. However, it does describe how to answer the question for a specific technology in a specific environment. Determining whether a technology is suitable for creating accessible content depends on several factors:

- Does the specification for the technology contain the necessary entities, properties, and relationships to enable the author to create conforming content?
- Do user agents, including assistive technology, support the technology and render the content in ways that are suitable for people with different disabilities?

There are a variety of ways in which technology specifications can support accessibility. The HTML specification [8] contains features such as the alt attribute for images or table markup that permit data cells and header cells to be associated. PDF 1.4 [9] includes logical structure and tagging, which defines the structure of the page in a way that is critical for accessibility. The Accessible Rich Internet Application (ARIA) specification [10] extends HTML and XHTML to expose critical information about the roles and states of elements of the page. SMIL supports captioning via synchronized text streams. Techniques for WCAG 2.0 [7] describe technology-specific methods of satisfying WCAG success criteria.

But having a technology that supports accessibility is not enough. User agents must also support these features to achieve access for people with disabilities. Not all user agents support all features, which was the source of various "Until user agents…" checkpoints in WCAG 1.0. Sometimes, user agents support features in their rendering, but do not expose the necessary information to assistive technology. The techniques in Techniques for WCAG 2.0 [7] include user agent notes that identify versions of user agents that have limitations or do not support the described technique. Once the accessibility features of a technology are supported by user agents and assistive technologies, it is said to be "accessibility-supported."

Of course, the author must use the accessibility-supported technology appropriately when creating content by following the WCAG 2.0 guidelines.

If authors use accessibility-supported Web technologies and follow the WCAG 2.0 guidelines, most users with disabilities should be able to access the content.

## 6. ACCESSIBILITY-SUPPORTED WEB TECHNOLOGIES

A technology is *accessibility-supported* if it contains features necessary to support accessibility and is supported by users' assistive technology as well as the accessibility features of browsers and other user agents. That is, the programs used to render the content make the content available to people with disabilities in the form and modality that meets their needs.

This raises the question of which user agents and assistive technology? This is a complex question, and the answer will vary by environment and by language.

- If the Web pages are internal to a closed community, such as a corporation or a school intranet, and will only be accessed by members of that community, then the technologies need only be supported by the user agents and assistive technology available in that community. That community may have policies about which user agents are supported, and may have the resources to ensure that everyone has access to current versions of user agents.
- If Web pages are available on the World Wide Web, the technology may need to be supported by a wide variety of user agents and assistive technology, including older versions, since users may not upgrade their software and hardware for many years.
- The language of the content must also be considered. The availability and level of support by user agents and assistive technologies in different languages varies widely.

If the requirements for user agent support are so broad as to require support by every version of assistive technology or browser, no matter how old or primitive, it will never be possible for new technologies to become accessibility-supported. But as the Web continues to evolve, it becomes less and less possible or practical to provide alternative versions that have equivalent functionality, and this will exclude people with disabilities from the mainstream. Furthermore, it will discourage Web authors using newer technologies from striving to satisfy WCAG 2.0, because they will never be able to conform.

However, if the requirements for user agent support are too narrow, pages that conform to WCAG 2.0 will be effectively inaccessible to people with disabilities who do not have the latest version of an assistive technology, for instance, or who do not use the platform on which the user agents run.

Determining whether a technology is accessibility-supported requires that an author understand the audience and what versions of what user agents or assistive technology are available to them. It also requires that the author understand the support provided by different user agents and assistive technology. This includes how different browsers and assistive technologies interact, as well as which technology each supports.

The information about the technology support of user agents and assistive technology is the same for all authors. It would benefit the Web community greatly if there were common shared sources of information about technologies and their accessibility support, rather than relying on each author to determine this individually.

## 7. USING TECHNOLOGIES THAT ARE NOT ACCESSIBILITY-SUPPORTED

But what about new technologies that don't have sufficient accessibility features or that are not yet supported by user agents and assistive technologies? Such technologies can be *used but not relied upon*. This is sometimes called *graceful degradation*, and it allows Web page authors flexibility in incorporating new technologies. It may permit the same Web page to be used in two different environments, in which different user agents are available and so different technologies are accessibility-supported. This provides a smoother path for new Web technologies to gain accessibility support.

All the information and functionality of the Web page must be available when user agents do not support the technologies that are not relied on. For instance, if CSS is *used but not relied on*, then the Web page must be usable in browsers with no support for CSS or with support disabled.

In addition, the Web page must also be usable when the user agent does support the technology that is not relied on. That is, the technology must not interfere with access to the page. For example, if Javascript is not relied on, but a Web page uses Javascript to animate part of the page in such a way that causes seizures, the page would not conform to WCAG 2.0.

Finally, WCAG 2.0 permits authors to provide conforming alternate versions of Web pages that do not conform. This may be necessary when it is not possible to use new technologies in a way that permits graceful degradation.

It may also be desirable to create customized pages that provide better support for people with a particular disability. For instance, a version customized for deaf users may include sign-language videos of the content. A version customized for users with visual impairments may provide a default format that omits images and uses large, plain fonts for text.

One of the ways to meet the needs of users with different cognitive, language, or learning disabilities is to provide alternate versions. Different versions may organize the information into smaller chunks, may avoid the use of specialized language, may use default formatting styles that make it easier to read the text, or may demonstrate all concepts in the text with graphics or video illustrations.

When alternate versions are provided, they must contain identical content and functionality, and it must be possible to locate the conforming version of the Web page from the non-conforming version.

## 8. OTHER TRENDS

In addition to the broad issues described above, WCAG 2.0 also wrestled with significant technical issues such as security, multimedia, and scripting.

### 8.1 Security

Security has emerged as a critical issue. Because malicious programs can pose as users, authors have had to devise various security techniques in order to validate that a user registering for access to a secure site is, in fact, a human being. One of the more popular techniques for validating users as humans is the Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) [11]. A CAPTCHA is typically an image with distorted text displayed over a patterned background; something that only a human being is expected to be able to decipher. Users are asked to enter the text string displayed in the CAPTCHA as part of the registration process for the website.

But for users with disabilities, CAPTCHAs present a barrier. Visually impaired users can't see them at all or can't see them well enough to enter the text string correctly. And users with cognitive disabilities may also have trouble entering the text string correctly. Adding a text alternative for an image is the typical method of making images accessible to visually impaired users. But in the case of CAPTCHAs, adding a text alternative would defeat the purpose of the image. If a screen reader can obtain the text string, so can a malicious program. So WCAG 2.0 allows an exception to the requirement for text alternatives in the case of CAPTCHAs and instead requires the CAPTCHA to also be presented in alternative modalities. For example, including an audio file that allows a blind user to hear the text in a visual CAPTCHA satisfies WCAG 2.0. Note that the wording is neutral in order to remain applicable to both visual and audio CAPTCHAs.

### 8.2 Multimedia

There has also been a dramatic rise in the amount of video content as high speed Internet connections have become the norm. Television networks can post their video news clips immediately after broadcasting them. Multimedia tutorials can be deployed to better illustrate something than can be done with text alone. And some of the most popular sites on the Web today even allow users to get in the game by publishing their own videos.

Producing professionally captioned and described videos is very expensive and time consuming. In addition, providing the audio descriptions requires very specialized skills that would be difficult, if not impossible, for most Web authors to acquire. In WCAG 2.0, captions are required for Level A conformance but audio descriptions are only required for Level AA conformance. At Level A, as an alternative to audio descriptions, it is acceptable to provide a full text alternative that includes sequenced text descriptions of all visual settings, actions, speakers, and non-speech sounds, and transcript of all dialog. If the video is interactive, the full text alternative must also provide a means for the interaction to occur at the appropriate time.

## 8.3 Scripting

And finally, scripting has become one of the most compelling new technologies to be widely utilized on websites.

With simple HTML, pages are displayed statically, one page at a time. Interactive links and submit buttons are used to obtain a new page or submit a form for processing which then causes a new page to be displayed. Each click on a link or submit button requires a round trip to the server to obtain the next. Users are forced to endure wait times which increase proportionally to the number of people accessing a particular Web server. The more popular the site, the longer the wait time.

With scripting, however, a Web developer can provide users with more of a desktop application "feel." For example, scripts can be used to selectively hide and display content to give the appearance of interactive menus. And with Asynchronous Javascript and XML (AJAX), even more robust GUI-type applications are possible and are becoming more commonplace. Word processors and spreadsheet applications can now be provided as a Web service without the maintenance issues of installing client applications. And this is only the beginning as developers are starting to unleash the power of the Web 2.0 environment.

This provides a better experience for users but also presents accessibility challenges. Even though browsers and assistive technologies now support scripts well, there are gaps in the technology. With scripting, developers can repurpose HTML elements to have the semantic meanings of graphical user interface controls. A common example of this is using a <div> element to define a menu that is hidden or displayed using a script. But a screen reader has no way of understanding that the <div> in this case is actually a menu. The only thing it can tell the user is that it is a list of links. In the case of a simple menu, this deficiency is not a significant problem as long as the "menu" is keyboard operable. But with more robust applications, the user needs more of an understanding than can currently be provided. The technology needs improvements to support accessibility; the Accessible Rich Internet Applications (ARIA) specifications [11] that extend HTML and XHTML address those gaps.

# 9. A CASE STUDY: ACCESSIBLE RICH INTERNET APPLICATIONS (ARIA)

How does WCAG 2.0 deal with the use of a new technology like ARIA and how does a new technology like ARIA help developers adopt WCAG 2.0?

ARIA is being developed to help address some specific accessibility issues that arise with the use of Javascript to create custom user interface controls. Developers of such scripting often find it difficult or impossible to satisfy several of the WCAG 2.0 success criteria, such as:

- *2.1.1 (Keyboard):* All functionality of the content is operable through a keyboard interface without requiring specific timings for individual keystrokes, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints.
- *4.1.2 (Name, Role, Value):* For all user interface components, the name and role can be programmatically determined, states, properties, and values that can be set by the user can be programmatically determined and programmatically set, and notification of changes to these items is available to user agents, including assistive technology.

With Javascript, developers can create new user interface controls in HTML and XHTML by using the <div> element and CSS and applying changes to the content dynamically. But no mechanism exists to identify the role of the <div> element, alert assistive technology when the element has keyboard focus, convey accessibility property information, or define what actions can be performed on the element.

The ARIA specification defines a mechanism for mapping such custom user interface controls to the accessibility frameworks on native operating system platforms. The WAI-ARIA Primer [13] provides a good overview of the problem and approach taken to solve it. In summary, ARIA addresses the following:

- *Role* by defining a role attribute that identifies the purpose of the control (role) and that can be mapped to platform accessibility programming interfaces (APIs). To improve usability a standard set of role attributes are defined to be used as landmarks for efficient keyboard navigation.
- *States and Properties* by defining state and property attributes that can be mapped to platform accessibility programming interfaces (APIs).
- *Keyboard Focus* by extending the TABINDEX feature to enable keyboard focus to be moved to any element.
- *Notification of changes* by defining events and user agent techniques required to support them.

ARIA can enable developers to better meet the needs of people with disabilities, so they will be able to enjoy the benefits provided by scripting just as their peers without disabilities do. However, is ARIA accessibility-supported? When can developers rely on it to satisfy WCAG 2.0?

Although the ARIA specification is not yet a W3C Recommendation, user agent and assistive technology support for ARIA is already emerging. In particular, Firefox 3 [14], and recent versions of Window-Eyes[15], JAWS[16], Orca[17], Fire Vox[18], and ZoomText [19] include support for some or all of the ARIA specification. Microsoft has announced plans to support ARIA in Internet Explorer 8 [20].

With ARIA and user agent and assistive technology support, Javascript, and AJAX in particular, will soon become accessibility-supported technologies. Developers who use them and conform to WCAG 2.0 will be creating Web 2.0 websites that people with disabilities can use.

## 10. MIGRATING FROM WCAG 1.0

Those who are familiar and comfortable with WCAG 1.0 will obviously be interested in understanding how it relates to the new WCAG 2.0 requirements.

In WCAG 1.0, guideline 5 (Create tables that transform gracefully), includes the following Priority 1 requirements:

- *5.1* For data tables, identify row and column headers.
- *5.2* For data tables that have two or more logical levels of row or column headers, use markup to associate data cells and header cells.

As worded, the WCAG 1.0 requirements are very specific to tables and to markup languages. But the reason for these requirements is that they programmatically identify the components of a table that can be perceived by sighted users from the visual presentation. Authors can achieve these visual effects without using table markup by using space to separate the columns and bold styling to distinguish column and row headers. This presentational technique is bad for accessibility *because* the table cannot be reliably rendered in an alternative modality by an assistive technology. But unless the author understands the rationale and principle behind a requirement, he or she might not know how to relate these requirements to non-HTML technologies.

In WCAG 2.0, these requirements are covered under the more general guideline 1.3. The guideline states the overall objective: “Create content that can be presented in different ways (for example simpler layout) without losing information or structure.”

And the success criterion 1.3.1 states the testable outcome: “Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text.” By wording the requirement in the standard as a general outcome that is to be achieved, it becomes a requirement that can be applied to any technology.

But how are authors supposed to translate this into action? They need techniques. There are at least 5 HTML techniques for tables in Techniques for WCAG 2.0 [7] that support this success criterion.

- Using table markup to present tabular information (HTML).
- Using caption elements to associate data table captions with data tables (HTML).
- Using the summary attribute of the table element to give an overview of data tables (HTML).
- Using the scope attribute to associate header cells and data cells in data tables (HTML).
- Using id and headers attributes to associate data cells with header cells in data tables.

The description of each technique provides examples and a test procedure specific to the technique. Note that authors are not required to implement all of these techniques. For example if the table doesn’t have a caption, the success criteria doesn’t require adding one. And ids and headers are not required for simple tables. The techniques provide guidance on applying them in the appropriate contexts.

Because techniques are informative, they can be modified as user agent and assistive technology support improves. And more can be added anytime for other technologies that support tables without having to change the standard itself.

Those who are familiar with WCAG 1.0 might want to start with the WCAG 2.0 Quick Reference [12] which lists the techniques after each success criterion. The quick reference can be customized by technology. The working group also plans to provide a mapping of WCAG 1.0 checkpoints to WCAG 2.0 success criteria.

## 11. POLICY

Many countries have implemented policies requiring conformance with WCAG 1.0. As policies take a long time to change, it is expected that WCAG 1.0 conformance will continue to be required for quite a while even though it is becoming more and more outdated. But at least two countries have begun to look at WCAG 2.0 in detail.

The United States Access Board formed an advisory committee, called the Telecommunications and Electronic Information Technology Advisory Committee (TEITAC) [21] in September 2006 charged with formulating a recommendation on refreshing the Section 508 and Section 255 [22] accessibility standards. The committee is very large and represents all stakeholders: government, consumers, industry, and invited experts. In the spirit of harmonization with accessibility policies in other countries, TEITAC also includes 4 international representatives: Canada, European Union, Japan, and Australia. The committee, which has almost finished their recommendation, took a detailed look at WCAG 2.0 for Web accessibility. The close review in a policy context was very beneficial to the WCAG process. As questions were raised and problems identified by the committee, the WCAG working group was able to adjust the wording to address those issues in most cases. While the committee recommendation doesn’t currently include all of the WCAG 2.0 Level A and AA requirements, it is harmonized with WCAG 2.0; that is, it does not contain any requirements for Web content that are unique to the United States.

In addition, China completed drafting their Web accessibility standards in 2007 and has based them on WCAG 2.0.

Several European countries have stated their intent to examine WCAG 2.0 when it is final but so far none have gone so far as to declare that they will move from WCAG 1.0 to 2.0. The European standards organizations ETSI and CEN have projects in process under EU Mandate 376 to develop standards and a certification process for IT accessibility. These projects will be looking closely at WCAG 2.0 and the work of the TEITAC.

## 12. CONCLUSION

The Web has changed dramatically during the development of WCAG 2.0, and shows every promise of continuing to evolve at a rapid pace. WCAG 2.0 consists of technology-neutral principles, guidelines, and success criteria that reflect properties of Web content that make it accessible to people with varying disabilities and combinations of disabilities. These guidelines and success criteria can be satisfied in a variety of ways in different Web formats and technologies. As the Web evolves, the guidelines can assist technology developers and authors in ensuring that people with disabilities continue to share in the benefits of the World Wide Web.

## 13. ACKNOWLEDGMENTS

Our thanks to the members of the WCAG Working Group.

## 14. REFERENCES

[1] Chisolm, W., Vanderheiden, G., Jacobs, I. *Web Content Accessibility Guidelines 1.0*. W3C Recommendation 5 May 1999.

[2] Jacobs, I., Gunderson, J., Hansen, E. *User Agent Accessibility Guidelines 1.0*. W3C Recommendation, 17 December 2002.

[3] Treviranus, J., McCathieNevile, C., Jacobs, I., Richards, J. *Authoring Tools Accessibility Guidelines 1.0*. W3C Recommendation, 3 February 2000.

[4] Architectural and Transportation Barriers Compliance Board. *Electronic and Information Technology Accessibility Standards; Final Rule*. Federal Register. CFR Part 1194. 21 December, 2000.

[5] Caldwell, B., Cooper, M., Guarino Reid, L., Vanderheiden, G. *Web Content Accessibility Guidelines 2.0*. W3C Working Draft, 11 December 2007.

[6] Caldwell, B., Cooper, M., Guarino Reid, L., Vanderheiden, G. *Understanding WCAG 2.0: A guide to understanding and implementing WCAG 2.0*. W3C Working Draft, 11 December 2007.

[7] Caldwell, B., Cooper, M., Guarino Reid, L., Vanderheiden, G. *Techniques for WCAG 2.0: Techniques and Failures for Web Content Accessibility Guidelines 2.0*. W3C Working Draft, 11 December 2007.

[8] Raggett, D., Le Hors, A., Jacobs, I. *HTML 4.01 Specification*, W3C Recommendation, 24 December 1999.

[9] *PDF Reference, Third Edition. Adobe Portable Document Format, Version 1.4*. Addison-Wesley.

[10] Seeman, L., Cooper, M., Schwerdtfeger, R., Pappas, L. *Accessible Rich Internet Applications (WAI-ARIA) Version 1.0* . W3C Working Draft, 4 February 2008.

[11] von Ahn, L., Blum, M., Hopper, N., and Langford, J. CAPTCHA: Using Hard AI Problems for Security. *EUROCRPYT 2003.*

[12] Vanderheiden, G., Guarino Reid, L, Caldwell, B., Lawton, S. *[Draft] WCAG 2.0 Quick Reference, A customizable list of WCAG 2.0 requirements (Success Criteria) and techniques*. 11 December 2007.

[13] Pappas, L., Schwerdtfeger, R., Cooper, M. *[Draft] WAI-ARIA Primer*, 4 February 2008.

[14] Mozilla Firefox project page. http://www.mozilla.org/products/firefox/.

[15] GWMicro Window-Eyes product page. http://www.gwmicro.com/Window-Eyes/

[16] Freedom Scientific JAWS® for Windows product page. http://www.freedomscientific.com/products/fs/jaws-product-page.asp

[17] Orca project page. http://www.gnome.org/projects/orca/

[18] Fire Vox product page. http://firevox.clcworld.net/

[19] AISquared ZoomText product page. http://www.aisquared.com/index.cfm/

[20] Internet Explorer 8 Beta 1 Whitepapers. http://code.msdn.microsoft.com/Release/ProjectReleases.aspx?ProjectName=ie8whitepapers&ReleaseId=564

[21] Architectural and Transportation Barriers Compliance Board. *Telecommunications Act Accessibility Guidelines; Electronic and Information Technology Accessibility Standards*. Federal Register, CFR Parts 1193 and 1194.

[22] *Telecommunications Act of 1996*. Pub.LA No. 104-104, 110 Stat. 56 (1996). Vol. 71, No. 129. July 6, 2006.