Authoring Tool Accessibility Guidelines 1.0 (working draft)

W3C Working Draft 22 October 1999

This version:
http://www.w3.org/WAI/AU/WAI-AUTOOLS-19991022
(plain text, HTML gzip tar archive, HTML zip archive, PostScript, PDF)
Latest version:
http://www.w3.org/WAI/AU/WAI-AUTOOLS
Previous version:
http://www.w3.org/WAI/AU/WAI-AUTOOLS-19991014

Editors:
Jutta Treviranus - ATRC, University of Toronto
Jan Richards - University of Toronto
Ian Jacobs - W3C
Charles McCathieNevile - W3C

Copyright © 1999 W3C® (MIT, INRIA, Keio), All Rights Reserved. W3C liability, trademark, document use and software licensing rules apply.

Abstract

This specification provides guidelines for Web authoring tool developers. Its purpose is two-fold: to assist developers in designing authoring tools that generate accessible Web content and to assist developers in creating an accessible authoring interface.

Authoring tool users ("authors") can be enabled, encouraged and assisted to create accessible Web content through prompts, alerts, checking and repair functions, help files and automated tools. It is equally important that all people can be the authors of Web content, rather than merely recipients. The tools used to create this information must therefore be accessible themselves. Adoption of these guidelines will contribute to the proliferation of Web content that can be read by a broader range of readers and in authoring tools that can be used by a broader range of authors.

This document is part of a series of accessibility documents published by the W3C Web Accessibility Initiative.

Status of this document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. The latest status of this document series is maintained at the W3C.

This is a W3C Working Draft for review by W3C Members and other interested parties. This draft follows the working group meeting on 20 October 1999 and reflects resolutions of that meeting. A log of changes between successive working drafts is available. For further information consult the minutes of Working Group Meetings.

This specification is a revision of the last call working draft dated 3 September 1999. The Working Group anticipates no further substantial changes to this specification before asking the director to give it the status of Proposed Recommendation.

Publication as a Working Draft does not imply endorsement by the W3C membership. This is still a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite W3C Working Drafts as other than "work in progress."

The goals of the WAI AU Working Group are discussed in the WAI AU charter.

Please send general comments about this document to the public mailing list: w3c-wai-au@w3.org, archived at http://lists.w3.org/Archives/Public/w3c-wai-au

A list of the current AU Working Group members is available.

A list of current W3C Recommendations and other technical documents can be found at http://www.w3.org/TR.

Table of Contents

Abstract 1
Status of this document 2
1. Introduction 4
1.1 How the Guidelines are organized. 5
1.2 Checkpoint Priorities 5
1.3 Conformance to these Guidelines 6
2. Guidelines 7
1. Support accessible authoring practices 7
2. Generate standard markup 7
3. Support the creation of accessible content 8
4. Provide methods of checking and correcting inaccessible content . .9
5. Integrate accessibility solutions into the overall "look and feel" . . 10
6. Promote accessibility in help and documentation 10
7. Ensure that the authoring tool is accessible to authors with disabilities 10
3. Glossary of Terms and Definitions 11
4. Acknowledgments 16
5. References 17

An appendix to this document [WAI-AUTOOLS-CHECKLIST] [p. 17] lists all checkpoints for convenient reference.

1. Introduction

The guidelines in this specification are designed to help authoring tool developers design authoring tools that can be used by people regardless of disability, and that produce accessible Web content. In these guidelines, the term authoring tool refers to the wide range of software used for creating Web content, including:

- Editing tools specifically designed to produce Web content (e.g., WYSIWYG HTML and XML editors);
- Tools that offer the option of saving material in a Web format (e.g., word processors or desktop publishing packages);
- Tools that translate documents into Web formats (e.g., filters to translate desktop publishing formats to HTML);
- Tools that produce multimedia, especially where it is intended for use on the Web (e.g., video production and editing suites, SMIL authoring packages);
- Tools for site management or site publication, including tools that generate Web sites dynamically from a database, on-the-fly conversion and Web site publishing tools;
- Tools for management of layout (e.g., CSS formatting tools).

The goals of this document can be stated as follows: that the authoring tool be accessible to authors regardless of disability, that it generate accessible content by default, and that it support and encourage the author in creating accessible content. Because most of the content of the Web is created using authoring tools, they play a critical role in ensuring the accessibility of the Web. Since the Web is both a means of receiving information and communicating information, it is important that both the Web content produced and the authoring tool itself be accessible.

This document provides guidelines for designing authoring tools that generate Web content which is accessible and that support and encourage authors to create such content. This is achieved by taking steps such as ensuring conformance to accessible standards (e.g., HTML 4.0), accessibility checking and correcting, prompting, appropriate documentation and help. For detailed information about what constitutes accessible content this specification relies on the Web Content Accessibility Guidelines [WAI-WEBCONTENT] [p. 17] . Similarly, rather than directly reproduce existing specifications that addresses general accessible software design the present specification relies on other sources. It does address accessible design considerations specific to Web authoring tools such as providing flexible editing views, navigation aids and access to display properties for authors.

A separate document, entitled "Techniques for Authoring Tool Accessibility" [WAI-AUTOOLS-TECHS] [p. 17] , provides suggestions and examples of how each checkpoint might be satisfied, It also includes references to other accessibility resources (such as platform-specific software accessibility guidelines) that provide additional information on how a tool may satisfy each checkpoint. Readers are strongly encouraged to become familiar with the techniques document. The Techniques provided in [WAI-AUTOOLS-TECHS] [p. 17] are suggestions for how to

satisfy the checkpoints, or where further information can be found. They are informative only, and other strategies may be used to meet the checkpoint as well as, or in place of, those discussed..

1.1 How the Guidelines are organized.

This document includes guidelines, which are general principles of accessible design. Each guideline includes:

- The guideline number;
- The statement of the guideline;
- The rationale behind the guideline;
- A list of checkpoint definitions.

The checkpoint definitions in each guideline specify requirements for authoring tools to follow the guideline. Each checkpoint definition includes:

- The checkpoint number;
- The statement of the checkpoint;
- The priority of the checkpoint;
- In some cases informative notes, clarifying examples, or cross references to related guidelines or checkpoints;
- A link to a section of [WAI-AUTOOLS-TECHS] [p. 17] where implementations and examples of the checkpoint are discussed;

Each checkpoint is intended to be specific enough that it can be verified, while being sufficiently general to allow developers the freedom to use the most appropriate strategies to meet the checkpoint.

An appendix to this specification [WAI-AUTOOLS-CHECKLIST] [p. 17] lists all checkpoints for convenient reference.

1.2 Checkpoint Priorities

Each checkpoint has a priority level. The priority level reflects the impact of the checkpoint in meeting the goals of this document. These goals are:

- That the authoring tool be accessible
- That the authoring tool generate accessible content by default
- That the authoring tool encourage the creation of accessible content

The three priority levels are assigned as follows:

[Priority 1]
: If the checkpoint is essential to meeting those goals

[Priority 2]
: If the checkpoint is important to meeting those goals

[Priority 3]
If the checkpoint is beneficial to meeting those goals
[Relative Priority]

Some checkpoints that refer to generating, authoring, or checking Web content have multiple priorities. The priority is dependent on the priority in the Web Content Accessibility Guidelines [WAI-WEBCONTENT] [p. 17] .

For example providing text equivalents [p. 12] for images and audio is a priority 1 requirement in [WAI-WEBCONTENT] [p. 17] since without it one or more groups will find it impossible to access the information. Therefore, it is a priority 1 requirement for the authoring tool to check for (4.1) or ask the author for (3.1) equivalent alternatives for these types of content. Expansion of abbreviations and acronyms with `ABBR` and `ACRONYM` elements by using the "`title`" attribute is a priority 3 in [WAI-WEBCONTENT] [p. 17] . Therefore it is only priority 3 for the authoring tool to check for (4.1) or ask the author for (3.2) this information.

- It is priority 1 to implement the checkpoint for content features that are a priority 1 requirement in [WAI-WEBCONTENT] [p. 17] .
- It is priority 2 to implement the checkpoint for content features that are a priority 2 requirement in [WAI-WEBCONTENT] [p. 17] .
- It is priority 3 to implement the checkpoint for content features that are a priority 3 requirement in [WAI-WEBCONTENT] [p. 17] .

1.3 Conformance to these Guidelines

This section defines three levels of conformance to this document:

- Conformance Level "A": all Priority 1 checkpoints are satisfied;
- Conformance Level "Double-A": all Priority 1 and 2 checkpoints are satisfied;
- Conformance Level "Triple-A": all Priority 1, 2, and 3 checkpoints are satisfied;

Note. Conformance levels are spelled out in text (e.g., "Double-A" rather than "AA") so they may be understood when rendered to speech.

Claims of conformance to this document must use one of the following two forms.

Form 1: Specify:

- The guidelines' title: "Authoring Tool Accessibility Guidelines 1.0 (working draft)"
- The guidelines' URI: http://www.w3.org/WAI/AU/WAI-AUTOOLS-19991022
- The conformance level satisfied: "A", "Double-A", or "Triple-A".
- The product covered by the claim (e.g., tool name and version number, upgrades or plug-ins required).
- The date of the claim.
- The checkpoints that are satisfied, and those that are considered not applicable

Example of Form 1: "MyAuthoringTool version 2.3 conforms to W3C's "Authoring Tool Accessibility Guidelines 1.0 (working draft)", available at http://www.w3.org/WAI/AU/WAI-AUTOOLS-19991022, level Double-A."

Form 2: Include, on each statement of conformance, one of three icons provided by W3C and link the icon to the appropriate W3C explanation of the claim.

[Editors' note: In the event this document becomes a Recommendation, by that date WAI will provide a set of three icons, for "A", "Double-A", or "Triple-A" conformance levels of "Authoring Tool Accessibility Guidelines 1.0 (working draft)", together with a stable URI to the W3C Web site for linking the icons to the W3C explanation of conformance claims.]

2. Guidelines

Guideline 1. Support accessible authoring practices

If the tool automatically generates markup, many authors will be unaware of the accessibility status of the final content unless they expend extra effort to make appropriate corrections by hand. Since many authors are unfamiliar with accessibility, the onus is on the authoring tool to generate accessible markup, and where appropriate, to guide the author in producing accessible content.

Many applications feature the ability to convert documents from other formats (e.g., Rich Text Format) into a markup format specifically intended for the Web such as HTML. Markup changes may also be made to facilitate efficient editing and manipulation. It is essential that these processes do not introduce inaccessible markup, or remove accessibility content, particularly since the markup changes are hidden from the author's view in many tools.

Checkpoints:

1.1 Ensure that the author can produce accessible content [p. 12] in the markup language(s) [p. 13] supported by the tool. [Priority 1]
1.2 Ensure that the tool preserves all accessibility information [p. ??] during authoring, transformations [p. 14] and conversions [p. 13] . [Priority 1]
1.3 Ensure that the tool generates markup that conforms to the W3C's Web Content Accessibility Guidelines [WAI-WEBCONTENT] [p. 17] . [Relative Priority]
1.4 Ensure that templates provided by the tool conform to the Web Content Accessibility Guidelines [WAI-WEBCONTENT] [p. 17] . [Relative Priority]

Guideline 2. Generate standard markup

Conformance with standards promotes interoperability and accessibility, by making it easier to create specialized user agents [p. 14] that address the needs of users with disabilities. In particular many assistive technologies used with browsers and multimedia players are only able to provide access to Web documents that use valid markup. Therefore valid markup is an essential aspect of authoring tool accessibility.

Where applicable use W3C Recommendations, which have been reviewed to ensure accessibility and interoperability. If there are no applicable W3C Recommendations, use a published standard that enables accessibility.

Checkpoints:

2.1 Use the latest versions of W3C Recommendations when they are available and appropriate for a task. [Priority 2]
W3C specifications have undergone review specifically to ensure that they do not compromise accessibility, and where possible they enhance it.
2.2 Ensure that the tool generates valid markup. [Priority 1]
This is necessary for user agents [p. 14] to be able to render Web content in a manner appropriate to a particular user's needs.
2.3 If markup generated by the tool does not conform to W3C specifications, inform the author. [Priority 3]

Guideline 3. Support the creation of accessible content

Well structured information, and equivalent alternative information [p. 12] are cornerstones of accessible design, allowing information to be presented in a way most appropriate for the needs of the user without constraining the creativity of the author. Generating equivalent information, such as textual alternatives for images and audio descriptions of video, can be one of the most challenging aspects of Web design. Automating the mechanics of this process, by prompting authors to include the relevant information at appropriate times, can greatly ease the burden for authors. Where such information can be mechanically determined and offered as a choice for the author (e.g., the function of icons in an automatically-generated navigation bar, or expansion of acronyms from a dictionary) the tool will assist the author. At the same time it can reinforce the need for such information and the author's role in ensuring that it is used appropriately in each instance.

Checkpoints:

3.1 Prompt [p. 14] the author to provide equivalent alternative information [p. 12] (e.g., captions, long descriptions of graphics). [Relative Priority]
3.2 Help the author create structured content and separate information from its presentation. [Relative Priority]
3.3 Ensure that prepackaged content conforms to [WAI-WEBCONTENT] [p. 17] . [Relative Priority]
For example include synchronized text and audio equivalents (such as video captions [p. 14]) with movies. Refer also to checkpoint 3.4.
3.4 Do not insert automatically generated or place-holder equivalent alternatives [p. 12] . [Priority 1]
For example, "search" may be appropriate alternative text for a graphic button linked to a search function, but the filename of an image should not be inserted as a default.

Note. Human-authored equivalent alternatives may be available for an object whose function is known with certainty. Refer also to checkpoint 3.5 and checkpoint 3.3.

3.5 Provide a mechanism to manage alternative information [p. 12] for multimedia objects, that retains and offers for editing pre-written or previously linked equivalent alternative information. [Priority 3]

Guideline 4. Provide methods of checking and correcting inaccessible content

Many authoring tools allow authors to create documents with little or no knowledge about the underlying markup. To ensure accessibility, authoring tools must be designed so that they can automatically identify inaccessible markup [p. 11] , and enable its correction even when the markup itself is hidden from the author.

In supporting the creation of accessible Web content, authoring tools should take into account differing authoring styles. In general, authors will prefer to be able to configure their tools to support their working style. Tools that allow such configuration can help authors to feel that accessible authoring is a natural practice guideline 5 rather than an intrusion on their normal work pattern. For example some users may prefer to be alerted to accessibilty problems [p. 11] when they occur, whereas others may prefer to perform a check at the end of an editing session. This is analogous to programming environments that allow users to decide whether to check for correct code during editing or at compile time.

Note. Validation of markup is an essential aspect of checking the accessibility of content.

Checkpoints:

4.1 Check for and alert [p. 12] the author to accessibility problems [p. 11] . [Relative Priority]

Note: Some accessibility problems cannot be detected automatically, and will require the user to make decisions.

4.2 Assist authors in correcting accessibility problems [p. 11] . [Relative Priority]

At a minimum, provide context-sensitive help with the accessibility checking required by 4.1

4.3 Allow the author to preserve markup not recognized by the tool. [Priority 2]

Note. The author may have included or imported markup that enhances accessibility but is not recognized by the tool.

4.4 Provide the author with a summary of the document's accessibility status. [Priority 3]

4.5 Allow the author to transform presentation markup that is misused to convey structure into structural markup, and to transform presentation markup that is stylistic into style sheets. [Priority 3]

Guideline 5. Integrate accessibility solutions into the overall "look and feel"

When a new feature is added to an existing software tool without proper integration, the result is often an obvious discontinuity. Differing color schemes, fonts, interaction styles and even application stability can be factors affecting user acceptance of the new feature. In addition, the relative prominence of different ways to achieve the same thing can be an important factor in which method an author chooses. Therefore it is important that creating accessible content is a natural process when using an authoring tool.

Checkpoints:

5.1 Ensure that functionalities related to accessible authoring practices [p. 12] are integrated into the tool. [Priority 2]
5.2 Ensure that [WAI-WEBCONTENT] [p. 17] Priority 1 accessible authoring practices [p. 12] are among the most obvious and easily initiated by the author. [Priority 2]

Guideline 6. Promote accessibility in help and documentation

The issues surrounding Web accessibility are often unknown to Web authors. Help and documentation should explain accessibility problems [p. 11] and solutions, with examples.

Checkpoints:

6.1 Document all features that promote the production of accessible content. [Priority 1]
6.2 Ensure that creating accessible content is a naturally integrated part of the documentation, including examples. [Priority 2]
6.3 In a dedicated section, document all features of the tool that promote the production of accessible content. [Priority 3]

Guideline 7. Ensure that the authoring tool is accessible to authors with disabilities

The authoring tool is a software program with standard user interface elements and as such should follow relevant user interface accessibility guidelines. In addition to applicable general interface accessibility guidelines there are interface design considerations that are specific to Web authoring tools.

One such consideration is that the author may need a different presentation to edit the Web content than the one they wish ultimately to be rendered. This implies display preferences that do not manifest themselves in the ultimate markup or style declarations.

Another consideration relates to the process of navigating and manipulating the document while authoring. Authoring Web content can require editing a large and complex document. In order to edit a document, the author must be able to locate and select specific elements, efficiently traverse the document, and quickly find and mark insertion points. Authors who use screen readers, refreshable braille displays, or screen magnifiers can make limited use (if at all) of visual artifacts that communicate the structure of the document and act as signposts when traversing it. Authors who use keyboard and mouse alternatives must make tiring repetitions of movement commands to navigate the document, and speed becomes an important issue. There are strategies that make it easier to navigate and manipulate a marked-up document. Using the structure of a Web document, the author can be given a view [p. 14] of it that provides the author with a good sense of the overall structure and facilitates navigating it. Developers should note that documentation, help files, and installation are part of the software and need to be available in an accessible [p. 12] form.

Checkpoints:

7.1 Use all applicable operating system and accessibility standards and conventions (Priority 1 for standards and conventions that are essential to accessibility, Priority 2 for those that are important to accessibility, Priority 3 for those that are beneficial to accessibility). [Priority 1]

> The techniques for this checkpoint include references to checklists and guidelines for a number of platforms and to general guidelines for accessible [p. 12] applications..

7.2 Allow the author to change the presentation within editing views [p. ??] without affecting the document markup. [Priority 1]

> This allows the author to edit the document according to their personal requirements, without changing the way the document is rendered when published.

7.3 Allow the author to edit all properties [p. 14] of each element [p. 13] and object in an accessible fashion. [Priority 1]

7.4 Ensure the editing view [p. 13] allows navigation via the structure of the document in an accessible fashion. [Priority 1]

7.5 Enable editing of the structure of the document in an accessible fashion. [Priority 2]

7.6 Allow the author to search within editing views [p. 13] . [Priority 2]

3. Glossary of Terms and Definitions

Access Barrier, Accessibility Problem, Inaccessible Authoring Practice, Inaccessible Markup

> These terms are used to mean markup or practices that do not meet (or produce content that does not meet) checkpoints of the Web Content Accessibility Guidelines [WAI-WEBCONTENT] [p. 17] .

Accessible, Accessibility

Within these guidelines, Accessible and Accessibility are used in the sense of being accessible to people regardless of disability.

To understand the accessibility issues relevant to authoring tool design, consider that many users may be creating documents in contexts very different from your own:

- They may not be able to see, hear, move, or may not be able to process some types of information easily or at all;
- They may have difficulty reading or comprehending text;
- They may not have or be able to use a keyboard or mouse;
- They may have a text-only display, or a small screen.

In addition, accessible design will benefit many people who do not have a physical disability but with similar needs. For example they may be working in a noisy environment and unable to hear, or need to use their eyes for another task, and be unable to view a screen. They may be using a small mobile device, with a small screen, no keyboard and no mouse. Both a piece of software such as an authoring tool and Web content can be accessible (or not).

Accessibility Information

Accessibility information is content, including information and markup, that is used to improve the accessibility of a document. Accessibility information includes, but is not limited to, equivalent alternative information [p. 12] .

Accessibility Solution, Accessible Authoring Practice

These terms refer to Authoring practices that improve the accessibility of content generated by the tool.

Alerts

Alerts notify the author of something, or mark something for the author's attention. They may or may not require author response.

Alternative Information, Equivalent Alternative

Certain types of content may not be accessible to all users, so alternative representations are used. For example, "text equivalents" are provided for all non-text elements because text can be flexibly output in ways that are usable to diverse populations. Specifically, text can be rendered aurally via synthesized speech for individuals who have visual or learning disabilities, tactually via braille for individuals who are blind, or as visually displayed text for individuals who are deaf or are non-disabled. Text equivalents for still images can be short ("Site Map Link") or long (e.g., "Figure 4 shows that the population of bacteria doubled approximately every twenty hours over the first one hundred hours, increasing from about 1000 per milliliter to about 32,000 per milliliter."). Text equivalents for audio clips are called "text transcripts". "Captions" are essential text equivalents for movie audio. Captions consist of a text transcript of the audio track of the movie (or other video presentation) that is synchronized with the video and audio tracks. Captions are generally displayed visually and benefit people who can see but are deaf, hard-of-hearing, or cannot hear the audio (e.g., in a noisy room). Another essential text equivalent for a movie is a

"collated text transcript," which combines (collates) caption text with text descriptions of video information (descriptions of the actions, body language, graphics, and scene changes of the video track). Collated text transcripts are essential for individuals who are deaf-blind and rely on braille for access to movies and other content. An essential non-text equivalent for movies is "auditory description" of the key visual elements of a presentation. The description is either a pre-recorded human voice or a synthesized voice (recorded or generated on the fly). The auditory description is synchronized with the audio track of the presentation, usually during natural pauses in the audio track. Auditory descriptions include information about actions, body language, graphics, and scene changes.

Attribute

in XML and HTML, an element may have any number of attributes. In the following example, the attributes of the beverage element are flavor, which has the value "lots", and colour, which has the value "red":

```
<beverage flavor="lots" colour="red">my favorite</beverage>
```

Some attributes are integral to document accessibility (e.g., the "alt", "title", and "longdesc" attributes in HTML

Authoring Tool

As used in this document, an *Authoring Tool* is any software that is used to generate content for publishing on the Web. Refer also to section 1.3 Scope of these guidelines.

Conversion Tool

A *Conversion Tool* is any application or application feature that allows content in some other format (proprietary or not) to be converted automatically into a particular markup language. This includes software whose primary function is to convert documents to a particular markup language as well as "save as HTML" (or other markup language) features in non-markup applications.

Document

A *document* is a series of elements that are defined by a language (e.g., HTML 4.0 or an XML application).

Editing View

A view [p. 14] provided by the authoring tool that allows editing. Some authoring tools will have several different types of view, and some allow views of several documents at once.

Element

An element is any identifiable object within a document, for example a character, word, image, paragraph or spreadsheet cell. In HTML and XML an element refers to a pair of tags and their content, or an "empty" tag - one that requires no closing tag or content.

Markup Language

The term *markup language* is used in this document to refer to the encoding language of a document, such as HTML, SVG, or MathML.

Prompts

Prompts are requests for user input, either information or a decision. Prompts require author response.For example, an "alt-text" entry field prominently displayed in an image insertion dialog would constitute a prompt. Prompts can be used to encourage authors to provide information needed to make the information accessible (such as alternative textual representations).

Property

A property is a piece of information about an element, for example structural information (e.g., it is item number 7 in a list, or plain text) or presentation information (e.g., that it is marked as bold, its font size is 14). In XML and HTML properties of an element include the name of the element (e.g., `IMG` or `DL`), the values of its attributes, and information associated by means of a style sheet. In a database, properties of a particular element may include values of the entry, and acceptable data types for that element.

Rendered Content

The *rendered content* is that which an element actually causes to be rendered by the user agent. This may differ from the element's structural content. For example, some elements cause external data to be rendered (e.g., the IMG element in HTML), and in some cases, browsers may render the value of an attribute (e.g., "alt", "title") in place of the element's content.

Transcripts

A transcript is a line by line record of all dialog and action within a video or audio clip.

Transformation

A process whereby one object is changed, according to a discrete set of rules, into another, equivalent, object. This includes any application or application feature that allows content which is marked up in a particular markup language to be transformed into another markup language, such as software that allows the author to change the DTD defined for the original document to another DTD.

User Agent

The term User Agent in this document refers to an application that is used to read web content, such as a browser, a plug-in for a particular media type, or a piece of assistive technology.

User Configurable Schedule

A user configurable schedule allows the user to determine the type of prompts and alerts that are used, including when they are presented.

Video Captions

Captions are essential text equivalents for movie audio. Captions consist of a text transcript of the audio track of the movie (or other video presentation) that is synchronized with the video and audio tracks. Captions are generally rendered visually and benefit people who can see but are deaf, hard-of-hearing, or cannot hear the audio.

Views

An authoring tool may offer several *views* of the same document. For instance, one view may show raw markup, a second may show a structured tree view, a third may show markup with rendered objects while a final view shows an

example of how the document may appear if it were to be rendered by a particular browser. A typical way to distinuguish views in a graphic environment is to place each in a separate window.

4. Acknowledgments

Many thanks to the following people who have contributed through review and comment: Jim Allan, Denis Anson, Kitch Barnicle, Kynn Bartlett, Harvey Bingham, Judy Brewer, Carl Brown, Dick Brown, Wendy Chisholm, Rob Cumming, Daniel Dardailler, Mark Day, BK Delong, Martin Dürst, Kelly Ford, Jamie Fox, Edna French, Sylvain Galineau, Al Gilman, Eric Hansen, Phill Jenkins, Len Kasday, Brian Kelly, Marja-Riitta Koivunen, Sho Kuwamoto, Jaap van Lelieveld, William Loughborough, Karen McCall, Charles Oppermann, Dave Pawson, Dave Poehlman, Bruce Roberts, Chris Ridpath, Gregory Rosmaita, Janina Sajka, Jim Thatcher, Irène Vatton, Gregg Vanderheiden, Pawan Vora, Jason White, and Lauren Wood.

5. References

For the latest version of any W3C specification please consult the list of W3C Technical Reports.

[WAI-AUTOOLS-CHECKLIST]
An appendix to this document lists all of the checkpoints, sorted by priority. The checklist is available in either tabular form (at http://www.w3.org/WAI/AU/WAI-AUTOOLS-19991022/checkpoint-table) or list form (at http://www.w3.org/WAI/AU/WAI-AUTOOLS-19991022/checkpoint-list).

[WAI-AUTOOLS-TECHS]
"Techniques for Authoring Tool Accessibility (working draft)," J. Treviranus, J. Richards, I. Jacobs, and C. McCathieNevile eds. The latest version is available at http://www.w3.org/WAI/AU/WAI-AUTOOLS-TECHS.

[WAI-WEBCONTENT]
"Web Content Accessibility Guidelines 1.0," W. Chisholm, G. Vanderheiden, and I. Jacobs, eds., 5 May 1999. This Recommendation is http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505. The latest version of the Web Content Accessibility Guidelines 1.0" is available at http://www.w3.org/TR/WAI-WEBCONTENT/.