[contents] [tabular checklist] [linear checklist]

Authoring Tool Accessibility Guidelines 1.0

W3C Recommendation 3 February 2000

This version:
http://www.w3.org/TR/2000/REC-ATAG10-20000203
(plain text, HTML gzip tar archive, HTML zip archive, PostScript, PDF)
Latest version:
http://www.w3.org/TR/ATAG10
Previous version:
http://www.w3.org/TR/1999/PR-WAI-AUTOOLS-19991026

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

Abstract

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

Authoring tools can enable, encourage, and assist users ("authors") in the creation of accessible Web content through prompts, alerts, checking and repair functions, help files and automated tools. It is just as important that all people be able to author content as it is for all people to have access to it. 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 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 (WAI).

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 document has been reviewed by W3C Members and other interested parties and has been endorsed by the Director as a W3C Recommendation. It is a stable document and may be used as reference material or cited as a normative reference from another document. W3C's role in making the Recommendation is to draw attention to the specification and to promote its widespread deployment. This enhances the functionality and interoperability of the Web.

A log of changes between successive Working Drafts is available.

For further information about Working Group decisions, please consult the minutes of AUWG Meetings.

This document has been produced by the Authoring Tool Accessibility Guidelines Working Group (AUWG) as part of the Web Accessibility Initiative (WAI). The goals of the Working Group are discussed in the AUWG charter.

Please send general comments about this document to the public mailing list: w3c-wai-au@w3.org (public archives).

The English version of this specification is the only normative version. Information about translations of this document is available at http://www.w3.org/WAI/AU/ATAG-TRANSLATIONS.

The list of known errors in this document is available at http://www.w3.org/WAI/AU/ATAG-ERRATA. Please report errors in this document to wai-atag-editor@w3.org.

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

Table of Contents

An appendix to this document [ATAG10-CHECKLIST] lists all checkpoints for convenient reference.

1. Introduction

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 transform documents into Web formats (e.g., filters to transform 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 automatically generate Web sites dynamically from a database, on-the-fly conversion tools, 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 produce 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.

To achieve these goals, authoring tool developers must take steps such as ensuring conformance to accessible standards (e.g., HTML 4), checking and correcting accessibility problems, prompting, and providing appropriate documentation and help. For detailed information about what constitutes accessible content, these guidelines rely on the Web Content Accessibility Guidelines 1.0 [WCAG10]. Similarly, rather than directly reproducing existing specifications that address general accessible software design, these guidelines rely on other sources. The present guidelines do address accessible design considerations specific to Web authoring tools such as providing flexible editing views, navigation aids and access to display properties for authors.

The principles set forth in these guidelines will benefit many people who do not have a disability but who have similar needs. This includes people who work in noisy or quiet environments where the use of sound is not practical, people who need to use their eyes for another task and are unable to view a screen, and people who use small mobile devices that have a small screen, no keyboard, and no mouse.

A separate document, entitled "Techniques for Authoring Tool Accessibility Guidelines 1.0" [ATAG10-TECHS], 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 as well as "Techniques for Web Content Accessibility Guidelines 1.0" [WCAG10-TECHS] and "Techniques for User Agent Accessibility Guidelines 1.0" [UAAG10-TECHS].

Note: The techniques in [ATAG10-TECHS] are informative examples only. Other strategies may be used to satisfy the checkpoints in addition to, or in place of, those discussed in [ATAG10-TECHS].

Note: Authoring tools that conform to this document will propagate accessible Web content and be useful to anyone regardless of disability. There will also be authoring tools that produce accessible content in favorable circumstances (e.g., a text editor used by a motivated author), or provide an accessible interface to authors with certain disabilities, but that do not conform to these guidelines.

1.1 How the Guidelines are organized

The seven guidelines in this document are general principles for 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 "Techniques for Authoring Tool Accessibility Guidelines 1.0" [ATAG10-TECHS] 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 satisfy it.

An appendix to this specification [ATAG10-CHECKLIST] 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 specification. These goals are:

- That the authoring tool be accessible;
- That the authoring tool produce accessible content by default;
- That the authoring tool encourage the creation of accessible content.

The priority levels are assigned as follows:

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

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

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

[Relative Priority]

Some checkpoints that refer to generating, authoring, or checking Web content have multiple priorities. The priority depends on the corresponding priority in the Web Content Accessibility Guidelines (WCAG) 1.0 [WCAG10].

- It is priority 1 to satisfy the checkpoint for content features that are a priority 1 requirement in WCAG 1.0.
- It is priority 2 to satisfy the checkpoint for content features that are a priority 2 requirement in WCAG 1.0.
- It is priority 3 to satisfy the checkpoint for content features that are a priority 3 requirement in WCAG 1.0.

For example:

- Providing *text equivalents* for images and audio is a priority 1 requirement in WCAG 1.0 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.

- Grouping links in navigation bars is a priority 3 in WCAG 1.0. Therefore, it is only priority 3 for the authoring tool to check for (4.1) or ask the author for (3.2) groups of links that are not grouped in the markup as a navigation mechanism.

When a checkpoint in this document refers to the WCAG 1.0 [WCAG10], only the WCAG 1.0 checkpoints that refer to content supported or automatically generated by the authoring tool apply. Some of the applicable WCAG 1.0 checkpoints may be satisfied automatically (without author participation) while others require human judgment and support from the tool in the form of prompts and documentation. Different tools may satisfy the same checkpoint differently.

The priority level for each checkpoint has been chosen based on the assumption that the author is a competent, but not necessarily expert, user of the authoring tool, and that the author has little or no knowledge of accessibility. For example, the author is not expected to have read all of the documentation, but is expected to know how to turn to the documentation for assistance.

1.3 Conformance to these Guidelines

This section explains how to make a valid claim that an authoring tool conforms to this document. Anyone may make a claim (e.g., vendors about their own products, third parties about those products, journalists about products, etc.). Claims may be published anywhere (e.g., on the Web or in product documentation).

Claimants are solely responsible for their claims and the use of the conformance icons. If the subject of the claim (i.e., the software) changes after the date of the claim, the claimant is responsible for updating the claim. Claimants are encouraged to conform to the most recent guidelines available.

Details about the conformance icons are provided on the Web (refer to [CONFORMANCE]).

Conformance levels

A conformance claim must indicate what conformance level is met:

- **Conformance Level "A"**: all Priority 1 checkpoints (including Relative Priority checkpoints) are satisfied.
- **Conformance Level "Double-A"**: all Priority 1 and 2 checkpoints (including Relative Priority checkpoints) are satisfied.
- **Conformance Level "Triple-A"**: all Priority 1, 2, and 3 checkpoints (including Relative Priority 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 as speech.

Well-formed conformance claims

A well-formed claim must include the following information:

1. The guidelines title/version: "Authoring Tool Accessibility Guidelines 1.0";
2. The URI of the guidelines: http://www.w3.org/TR/2000/REC-ATAG10-20000203;
3. The conformance level satisfied: "A", "Double-A", or "Triple-A";
4. The version number and operating system of the software covered by the claim. Also indicate whether any upgrades or plug-ins are required;
5. The date of the claim;
6. The checkpoints of the chosen conformance level considered not applicable. Claimants should use the checklist [ATAG10-CHECKLIST] for this purpose.

This information may be provided in text or metadata markup (e.g., using the Resource Description Framework (RDF) [RDF10] and an RDF schema designed for WAI conformance claims). All content in the claim must be accessible according to the Web Content Accessibility Guidelines 1.0 [WCAG10].

Here is an example of a claim expressed in HTML:

```
<p>MyAuthoringTool version 2.3 on MyOperatingSystem conforms
to <abbr title="the World Wide Web Consortium">W3C</abbr>'s
"Authoring Tool Accessibility Guidelines 1.0", available at
http://www.w3.org/TR/2000/REC-ATAG10-20000203, level Double-
A. Details of this claim are provided at <a
href="http://somewhere.com/details">
http://somewhere.com/details</a>.</p>
```

Validity of a claim

A conformance claim is valid for a given conformance level if:

1. The claim is well-formed, and
2. The authoring tool satisfies all the checkpoints for that level.

Claimants (or relevant assuring parties) are responsible for the validity of a claim. As of the publication of this document, W3C does not act as an assuring party, but it may do so in the future, or establish recommendations for assuring parties.

Claimants are expected to modify or retract a claim if it may be demonstrated that the claim is not valid. Please note that it is not currently possible to validate claims completely automatically.

Conformance Icons

As part of a conformance claim, people may use a conformance icon on a Web site, on product packaging, in documentation, etc. Each conformance icon (chosen according to the appropriate conformance level) must link to the W3C explanation of the icon. The appearance of a conformance icon does not imply that W3C has reviewed or validated the claim. An icon must be accompanied by a well-formed claim.

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 review it and make appropriate corrections by hand. Since many authors are unfamiliar with accessibility, authoring tools are responsible for automatically generating accessible markup, and where appropriate, for guiding 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 when a tool hides the markup changes from the author's view.

Checkpoints:

1.1 Ensure that the author can produce accessible content in the markup language(s) supported by the tool. [Priority 1]
Techniques for checkpoint 1.1

1.2 Ensure that the tool preserves all accessibility information during authoring, transformations, and conversions. [Priority 1]
Techniques for checkpoint 1.2

1.3 Ensure that when the tool automatically generates markup it conforms to the W3C's Web Content Accessibility Guidelines 1.0 [WCAG10]. [Relative Priority]
Techniques for checkpoint 1.3

1.4 Ensure that templates provided by the tool conform to the Web Content Accessibility Guidelines 1.0 [WCAG10]. [Relative Priority]
Techniques for checkpoint 1.4

Guideline 2. Generate standard markup.

Conformance with standards promotes interoperability and accessibility by making it easier to create specialized *user agents* 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.
> Techniques for checkpoint 2.1

2.2 Ensure that the tool automatically generates valid markup. [Priority 1]

> This is necessary for *user agents* to be able to render Web content in a manner appropriate to a particular user's needs.
> Techniques for checkpoint 2.2

2.3 If markup produced by the tool does not conform to W3C specifications, inform the author. [Priority 3]

> Techniques for checkpoint 2.3

Guideline 3. Support the creation of accessible content.

Well-structured information and *equivalent alternative information* 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. Yet producing equivalent information, such as text alternatives for images and auditory descriptions of video, can be one of the most challenging aspects of Web design, and authoring tool developers should attempt to facilitate and automate the mechanics of this process. For example, prompting authors to include equivalent alternative information such as *text equivalents*, *captions*, and *auditory descriptions* 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 can assist the author. At the same time, the tool 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 the author to provide equivalent alternative information (e.g., captions, auditory descriptions, and collated text transcripts for video). [Relative Priority]

> **Note:** Some checkpoints in the Web Content Accessibility Guidelines 1.0 [WCAG10] may not apply.
> Techniques for checkpoint 3.1

3.2 Help the author create structured content and separate information from its presentation. [Relative Priority]

> **Note:** Some checkpoints in Web Content Accessibility Guidelines 1.0 [WCAG10] may not apply.
>
> Techniques for checkpoint 3.2

3.3 Ensure that prepackaged content conforms to the Web Content Accessibility Guidelines 1.0 [WCAG10]. [Relative Priority]

> For example, include *captions*, an *auditory description*, and a *collated text transcript* with prepackaged movies. Refer also to checkpoint 3.4.
>
> Techniques for checkpoint 3.3

3.4 Do not automatically generate equivalent alternatives. Do not reuse previously authored alternatives without author confirmation, except when the function is known with certainty. [Priority 1]

> For example, *prompt* the author for a *text equivalent* of an image. If the author has already provided a text equivalent for the same image used in another document, offer to reuse that text and prompt the author for confirmation. If the tool automatically generates a "Search" icon, it would be appropriate to automatically reuse the previously authored text equivalent for that icon. Refer also to checkpoint 3.3 and checkpoint 3.5.
>
> **Note:** Human-authored equivalent alternatives may be available for an object (for example, through checkpoint 3.5 and/or checkpoint 3.3). It is appropriate for the tool to offer these to the author as defaults.
>
> Techniques for checkpoint 3.4

3.5 Provide functionality for managing, editing, and reusing alternative equivalents for multimedia objects. [Priority 3]

> **Note:** These alternative equivalents may be packaged with the tool, written by the author, retrieved from the Web, etc.
>
> Techniques for checkpoint 3.5

Guideline 4. Provide ways 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 (where possible, automatically) identify *inaccessible markup*, and enable its correction even when the markup itself is hidden from the author.

Authoring tool support for the creation of accessible Web content should account for different authoring styles. Authors who can configure the tool's accessibility features to support their regular work patterns are more likely to accept accessible authoring practices (refer to guideline 5). For example, some authors may prefer to be alerted to *accessibility problems* 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 compilation.

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

Checkpoints:

4.1 Check for and inform the author of accessibility problems. [Relative Priority]

> **Note:** Accessibility problems should be detected automatically where possible. Where this is not possible, the tool may need to *prompt* the author to make decisions or to manually check for certain types of problems.
> Techniques for checkpoint 4.1

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

> At a minimum, provide context-sensitive help with the accessibility checking required by checkpoint 4.1
> Techniques for checkpoint 4.2

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.
> Techniques for checkpoint 4.3

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

> Techniques for checkpoint 4.4

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

> Techniques for checkpoint 4.5

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 software stability can be factors affecting author acceptance of the new feature. In addition, the relative prominence of different ways to accomplish the same task can influence which one the author chooses. Therefore, it is important that creating accessible content be a natural process when using an authoring tool.

Checkpoints:

5.1 Ensure that functionality related to accessible authoring practices is naturally integrated into the overall look and feel of the tool. [Priority 2]

> Techniques for checkpoint 5.1

5.2 Ensure that accessible authoring practices supporting Web

Content Accessibility Guidelines 1.0 [WCAG10] Priority 1 checkpoints are among the most obvious and easily initiated by the author. [Priority 2]

Techniques for checkpoint 5.2

Guideline 6. Promote accessibility in help and documentation.

Web authors may not be familiar with accessibility issues that arise when creating Web content. Therefore, help and documentation must include explanations of *accessibility problems*, and should demonstrate solutions with examples.

Checkpoints:

6.1 Document all features that promote the production of accessible content. [Priority 1]

Techniques for checkpoint 6.1

6.2 Ensure that creating accessible content is a naturally integrated part of the documentation, including examples. [Priority 2]

Techniques for checkpoint 6.2

6.3 In a dedicated section, document all features of the tool that promote the production of accessible content. [Priority 3]

Techniques for checkpoint 6.3

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 must be designed according to relevant user interface accessibility guidelines. When custom interface components are created, it is essential that they be accessible through the standard access mechanisms for the relevant platform so that assistive technologies can be used with them.

Some additional user interface design considerations apply specifically to *Web authoring tools.* For instance, authoring tools must ensure that the author can edit (in an *editing view*) using one set of stylistic preferences and publish using different styles. Authors with low vision may need large text when editing but want to publish with a smaller default text size. The style preferences of the editing view must not affect the markup of the published document.

Authoring tools must also ensure that the author can navigate a document efficiently while editing, regardless of disability. Authors who use screen readers, refreshable braille displays, or screen magnifiers can make limited use (if at all) of graphical artifacts that communicate the structure of the document and act as signposts when traversing it. Authors who cannot use a mouse (e.g., people with physical disabilities or who are blind) must use the slow and tiring process of moving one step at a time through the document to access the desired content, unless more efficient navigation methods are available.

Authoring tools should therefore provide an *editing view* that conveys a sense of the overall structure and allows structured navigation.

Note: Documentation, help files, and installation are part of the software and need to be available in an *accessible* 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).

The techniques for this checkpoint include references to checklists and guidelines for a number of platforms and to general guidelines for *accessible* applications.

Techniques for checkpoint 7.1

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

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

Techniques for checkpoint 7.2

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

Techniques for checkpoint 7.3

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

Techniques for checkpoint 7.4

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

Techniques for checkpoint 7.5

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

Techniques for checkpoint 7.6

3. Glossary of Terms and Definitions

***Accessibility* (Also: *Accessible*)**

Within these guidelines, "accessible Web content" and "accessible authoring tool" mean that the content and tool can be used by people regardless of disability.

To understand the accessibility issues relevant to authoring tool design, consider that many authors may be creating content 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.

Accessible design will benefit people in these different authoring scenarios and also many people who do not have a physical disability but who have similar needs. For example, someone may be working in a noisy environment and thus require an alternative representation of audio information. Similarly, someone may be working in an eyes-busy environment and thus require an audio equivalent to information they cannot view. Users of small mobile devices (with small screens, no keyboard, and no mouse) have similar functional needs as some users with disabilities.

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*.

Accessibility Problem (Also: *Inaccessible Markup*)

Inaccessible Web content or authoring tools cannot be used by some people with disabilities. The Web Content Accessibility Guidelines 1.0 [WCAG10] describes how to create accessible Web content.

Accessible Authoring Practice

"Accessible authoring practices" improve the accessibility of Web content. Both authors and tools engage in accessible authoring practices. For example, authors write clearly, structure their content, and provide navigation aids. Tools automatically generate valid markup and assist authors in providing and managing appropriate equivalent alternatives.

Alert

An "alert" draws the author's attention to an event or situation. It may require a response from the author.

Alternative Information (Also: *Equivalent Alternative*)

Content is "equivalent" to other content when both fulfill essentially the same function or purpose upon presentation to the user. Equivalent alternatives play an important role in accessible authoring practices since certain types of content may not be accessible to all users (e.g., video, images, audio, etc.). Authors are encouraged to provide text equivalents for non-text content since text may be rendered as synthesized speech for individuals who have visual or learning disabilities, as braille for individuals who are blind, or as graphical text for individuals who are deaf or do not have a disability. For more information about equivalent alternatives, please refer to the Web Content Accessibility Guidelines WCAG 1.0 [WCAG10].

Attribute

This document uses the term "attribute" as used in SGML and XML ([XML]): Element types may be defined as having any number of attributes. Some attributes are integral to the accessibility of content (e.g., the `"alt"`, `"title"`, and `"longdesc"` attributes in HTML).

Auditory Description

An "auditory description" provides information about actions, body language, graphics, and scene changes in a video. Auditory descriptions are commonly used by people who are blind or have low vision, although they may also be used as a low-bandwidth equivalent on the Web. An auditory description is either a pre-recorded human voice or a synthesized voice (recorded or automatically generated in real time). The auditory description must be synchronized with the auditory track of a video presentation, usually during natural pauses in the auditory track.

Authoring Tool

An "authoring tool" is any software that is used to produce content for publishing on the Web. Authoring tools include:

- 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 transform documents into Web formats (e.g., filters to transform 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 automatically 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).

Captions

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

Conversion Tool

A "conversion tool" is any application or application feature (e.g., "Save as HTML") that transforms convent in one format to another format (such as a markup language).

Check for

As used in checkpoint 4.1, "check for" can refer to three types of checking:

1. In some instances, an authoring tool will be able to check for accessibility problems automatically. For example, checking for validity (checkpoint 2.2) or testing whether an image is the only content of a link.
2. In some cases, the tool will be able to "suspect" or "guess" that

there is a problem, but will need confirmation from the author. For example, in making sure that a sensible reading order is preserved a tool can present a linearized version of a page to the author.

3. In some cases, a tool must rely mostly on the author, and can only ask the author to check. For example, the tool may prompt the author to verify that equivalent alternatives for multimedia are appropriate. This is the minimal standard to be satisfied. Subtle, rather than extensive, prompting is more likely to be effective in encouraging the author to verify accessibility where it cannot be done automatically.

Document

A "document" is a series of elements that are defined by a *markup language* (e.g., HTML 4 or an XML application).

Editing View

An "editing view" is a *view* provided by the authoring tool that allows editing.

Element

An "element" is any identifiable object within a document, for example, a character, word, image, paragraph or spreadsheet cell. In [HTML4] 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.

Inform

To "inform" is to make the author aware of an event or situation through *alert*, *prompt*, sound, flash, or other means.

Markup Language

Authors encode information using a "markup language" such as HTML [HTML4], SVG [SVG], or MathML [MATHML].

Presentation Markup

"Presentation markup" is *markup language* that encodes information about the desired presentation or layout of the content. For example, Cascading Style Sheets ([CSS1], [CSS2]) can be used to control fonts, colors, aural rendering, and graphical positioning. Presentation markup should not be used in place of *structural markup* to convey structure. For example, authors should mark up lists in HTML with proper list markup and style them with CSS (e.g., to control spacing, bullets, numbering, etc.). Authors should not use other CSS or HTML incorrectly to lay out content graphically so that it resembles a list.

Prompt

A "prompt" is a request for author input, either information or a decision. A prompt requires author response. For example, a *text equivalent* 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 content accessible (such as *alternative text equivalents*).

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 type 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 entry.

Structural Markup

"Structural markup" is *markup language* that encodes information about the structural role of elements of the content. For example, headings, sections, members of a list, and components of a complex diagram can be identified using structural markup. Structural markup should not be used incorrectly to control presentation or layout. For example, authors should not use the `BLOCKQUOTE` element in HTML [HTML4] to achieve an indentation visual layout effect. Structural markup should be used correctly to communicate the roles of the elements of the content and *presentation markup* should be used separately to control the presentation and layout.

Transcript

A "transcript" is a text representation of sounds in an audio clip or an auditory track of a multimedia presentation. A "collated text transcript" for a video combines (collates) caption text with text descriptions of video information (descriptions of the actions, body language, graphics, and scene changes of the visual track). Collated text transcripts are essential for individuals who are deaf-blind and rely on braille for access to movies and other content.

Transformation

A "transformation" is a process that changes a document or object into another, equivalent, object according to a discrete set of rules. This includes *conversion tools*, software that allows the author to change the DTD defined for the original document to another DTD, and the ability to change the markup of lists and convert them into tables.

User Agent

A "user agent" is software that retrieves and renders Web content. User agents include browsers, plug-ins for a particular media type, and some assistive technologies.

View

Authoring tools may render the same content in a variety of ways; each rendering is called a "view." Some authoring tools will have several different types of view, and some allow views of several documents at once. For instance, one view may show raw markup, a second may show a structured tree, 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 distinguish 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, Aaron Cohen, Rob Cumming, Daniel Dardailler, Mark Day, BK Delong, Martin Dürst, Kelly Ford, Jamie Fox, Edna French, Sylvain Galineau, Al Gilman, Jon Gunderson, Eric Hansen, Phill Jenkins, Len Kasday, Brian Kelly, Marja-Riitta Koivunen, Sho Kuwamoto, Jaap van Lelieveld, Susan Lesch, William Loughborough, Greg Lowney, Karen McCall, Thierry Michel, Charles Oppermann, Dave Pawson, Dave Poehlman, Loretta Reid, Bruce Roberts, Chris Ridpath, Gregory Rosmaita, Bridie Saccocio, Janina Sajka, John Slatin, 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 at http://www.w3.org/TR.

[ATAG10-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/TR/2000/REC-ATAG10-20000203/atag10-chktable) or list form (at http://www.w3.org/TR/2000/REC-ATAG10-20000203/atag10-chklist).

[ATAG10-TECHS]
"Techniques for Authoring Tool Accessibility Guidelines 1.0," J. Treviranus, J. Richards, I. Jacobs, and C. McCathieNevile eds. The latest version is available at http://www.w3.org/TR/ATAG10-TECHS.

[CONFORMANCE]
"Conformance icons for ATAG 1.0." Information about ATAG 1.0 conformance icons is available at http://www.w3.org/WAI/ATAG10-Conformance.

[CSS1]
"CSS, level 1 Recommendation," B. Bos and H. Wium Lie, eds., 17 December 1996, revised 11 January 1999. This CSS1 Recommendation is http://www.w3.org/TR/1999/REC-CSS1-19990111. The latest version of CSS1 is available at http://www.w3.org/TR/REC-CSS1. **Note:** CSS1 has been superseded by CSS2. Tools should implement the CSS2 cascade.

[CSS2]

"CSS, level 2 Recommendation," B. Bos, H. Wium Lie, C. Lilley, and I. Jacobs, eds., 12 May 1998. This CSS2 Recommendation is http://www.w3.org/TR/1998/REC-CSS2-19980512. The latest version of CSS2 is available at http://www.w3.org/TR/REC-CSS2.

[HTML4]

"HTML 4.01 Recommendation," D. Raggett, A. Le Hors, and I. Jacobs, eds., 24 December 1999. This HTML 4.01 Recommendation is http://www.w3.org/TR/1999/REC-html401-19991224. The latest version of HTML 4 is available at http://www.w3.org/TR/html4.

[MATHML]

"Mathematical Markup Language," P. Ion and R. Miner, eds., 7 April 1998, revised 7 July 1999. This MathML 1.0 Recommendation is http://www.w3.org/TR/1998/REC-MathML-19990707. The latest version of MathML 1.0 is available at http://www.w3.org/TR/REC-MathML.

[RDF10]

"Resource Description Framework (RDF) Model and Syntax Specification," O. Lassila, R. Swick, eds. The 22 February 1999 Recommendation is http://www.w3.org/TR/1999/REC-rdf-syntax-19990222. The latest version of RDF 1.0 is available at http://www.w3.org/TR/REC-rdf-syntax.

[SVG]

"Scalable Vector Graphics (SVG) 1.0 Specification (Working Draft)," J. Ferraiolo, ed. The latest version of the SVG specification is available at http://www.w3.org/TR/SVG.

[UAAG10-TECHS]

"Techniques for User Agent Accessibility Guidelines 1.0," J. Gunderson, and I. Jacobs, eds. The latest version of Techniques for User Agent Accessibility Guidelines 1.0 is available at http://www.w3.org/TR/UAAG10-TECHS/.

[WCAG10]

"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/WCAG10/.

[WCAG10-TECHS]

"Techniques for Web Content Accessibility Guidelines 1.0," W. Chisholm, G. Vanderheiden, and I. Jacobs, eds. The latest version of Techniques for Web Content Accessibility Guidelines 1.0 is available at http://www.w3.org/TR/WCAG10-TECHS/.

[XML]

"The Extensible Markup Language (XML) 1.0," T. Bray, J. Paoli, C. M. Sperberg-McQueen, eds., 10 February 1998. This XML 1.0 Recommendation is http://www.w3.org/TR/1998/REC-xml-19980210. The latest version of the XML specification is available at http://www.w3.org/TR/REC-xml.

[contents] [tabular checklist] [linear checklist]