Initial Audio Characteristics for XSL

Christopher R. Maden
Senior Tools Specialist
O'Reilly & Associates, Inc.
90 Sherman Street
Cambridge, MA USA 02140
crism@oreilly.com

World-Wide Web Consortium XSL Working Group

Revision 0.1 7 May 1998 Initial draft.

Abstract: In keeping with the W3C's mission of accessibility, XSL must make an effort to make accessible documents as easy to create as possible. The more difficult accessibility is to achieve, the fewer documents will be accessible, as authors will not perform extra work to benefit what is often perceived as an insignificant minority. I propose adding audio characteristics derived from Aural Cascading Style Sheets to the otherwise visual flow objects in the July 1998 draft of XSL.

Introduction

One of the greatest problems that faces accessibility advocates on the Web today is the perception by many Web designers that accessibility requires extra work on their part, or even that it requires lower-quality designs for their sighted audience. Because of this perception, I feel very strongly that XSL must allow "easy accessibility" from the very beginning of its design.

This must be done in a way that does not impede future capabilities enabling high-quality audio stylesheets, high-quality visual stylesheets, or high-quality multimedia stylesheets. The goal is that careful designers should be able to create works of beauty in any medium, but that a lazy designer's visual stylesheet should have a well-defined audio analog.

It is arguable that enabling easy accessibility will justify continued laziness on the part of designers. However, empirical evidence on the Web indicates that most Web designers will perform the minimal necessary work to achieve a desired effect for most of the audience. Recent application of the Americans with Disabilities Act to Internet information resources is beginning to have an effect on this, but this may only affect public resources such as government information pages. Unless accessibility becomes much easier to achieve, the visually impaired will continue to be second-class citizens of the Web, and if accessibility itself is a second-class priority of the XSL design effort, that will only worsen the situation.

The W3C working draft on Aural Cascading Style Sheets (ACSS)[1] defines a set of audio characteristics (or properties) for use with CSS level 1.[2] Some of these are still preliminary, and the ACSS draft contains unanswered questions about their exact nature. Other properties seem robust and simple enough to be included in the initial XSL draft.

Applicability of Audio Characteristics

Most of the audio characteristics alter the manner in which the content of an element is read, while others provide background noise or sound effects. The applicability of those affecting textual presentation seems obvious, while the difference between XSL and CSS has an interesting effect on sound effects.

Characteristics Affecting Textual Delivery

These characteristics are only applicable to flow objects whose content is a series of character flow objects. The characteristics that inherit can be meaningful when applied to larger flow objects, naturally (*e.g.*, an entire series of paragraphs could have a slightly higher volume).

It is interesting to note that while visual formatting can be a function of individual character flow objects (with some interaction between neighboring characters in certain scripts), speech rendering is a function of a series of characters, possibly across flow object boundaries, necessitating a slight change in terminology.

These characteristics are:

- azimuth
- elevation
- pitch
- pitch-range
- richness
- speak-numeral
- speak-punctuation
- speech-rate
- stress
- voice-family
- volume

Backgrounds and Cues

ACSS provides properties that generate sound effects before, during, or after the text of an element is read. The ability of XSL to generate multiple flow objects for a single element removes the need for multiple events to be characteristics, but it introduces a new problem.

Flow objects that provide only a sound effect can be thought of as flow objects of some length with no content, but with a background sound. However, ACSS allows the sound effect to be specified by URI. There are a few ways we could provide this functionality in XSL.

The first is to provide a set of characteristics roughly equivalent to the property and keywords of ACSS's play-during property. One characteristic would take a URL as its value, another would specify whether or not to repeat the sound effect, and a third would state whether the sound effect should combine with or replace any background currently in place.

The other would be to have an audio flow object. This object would have as a URI as a characteristic, along with controls such as volume. The flow object itself would create a single iteration of the specified sound; repetition of the sound would be handled by another flow object "tiling" the audio flow object for its own duration.

The second option is cleaner: it can be used for sound effect before or after (or even during) an element's audio rendering, and could possibly be generalized to video or other non-textual events, or serve as a template for other specialized flow objects. Sound effect cues under the first model would need to be silent flow objects of a certain duration with a background sound, since the ACSS specification states that the background sound shall be truncated to the length of the element being rendered. The cue flow object would need to know the length of the sound effect in order to provide clearance, which seems unnecessarily cumbersome.

On the other hand, providing clearance for the background sounds nicely provides a mechanism for pauses; see the discussion of debatable characteristics.

These characteristics affect the duration of an element's rendering, by events before or after, or continual effects:

- pause
- pause-after
- pause-before
- play-during

Audio Characteristic Definitions

Each characteristic specified as a property in ACSS is discussed below. Two are proposed as (relatively) uncontroversial candidates for inclusion in the July draft of XSL; eight others are proposed as more controversial candidates, with pros and cons discussed. Five more are considered inappropriate for inclusion, and five are worth consideration, but are too immature for the July draft.

Characteristics Retained from ACSS

volume

The volume of a spoken element seems like an obvious and uncontroversial characteristic. When applied to any flow object with character flow object children, it governs the volume used to speak the words composed of those characters.

ACSS specifies that this characteristic should take a percentage, where 0% is the quietest audible volume, and 100% is the loudest comfortable volume. It also specifies five named volumes, plus "silent", which means that the element's rendering occupies sufficient time for it to be spoken, but that no sound is actually produced.

The default value for an XSL flow object with no explicit or inherited specification should be 50%.

speak-punctuation

This characteristic governs whether punctuation within the rendered element should be spoken specifically, or merely used to guide inflection of the text. In code listings or grammar texts where the exact punctuation is important, this is a useful characteristic.

In ACSS, this characteristic takes the values “code” or “none”, with a default value of “none”. I feel that a Boolean value is more appropriate, and that the default should be false.

Debatable Characteristics

pause-before, pause-after, pause

Given XSL's ability to generate multiple flow objects from a single element, pause-before and pause-after are not strictly necessary if pause is present. However, by analogy with space-before and space-after in visual stylesheets, they are conceptually associated with the element being rendered, and probably deserve their own characteristics.

In addition, providing these characteristics makes specification of audio behavior easier for visual flow objects, as that behavior can be specified as defaults on these characteristics; otherwise the specification must be left to prose description. See the section on user agent behavior.

The pause characteristic itself seems meritorious, to provide sonic whitespace. The reason it is listed as debatable is that it depends on the outcome of an audio flow object; if one is created, the pause could be presented as an audio flow object of a certain duration with no background.

play-during

As in the discussion about audio flow objects, play-during could be handled as a URI directly, or its value could be an audio flow object. All of this assumes that we think that it is important enough to include in the initial draft.

azimuth, elevation

These are simple and unambiguous; they specify the location from which the sound of an element should appear to come. The question is whether they are too simple and might block the development of a clean solution for more complex audio location capabilities, such as a speaker who moves while speaking.

speech-rate

This controls the rate at which an element is presented to the listener. As with location characteristics, the question is whether this characteristic represents an over-simplification that might be clutter in the face of a more complete solution, such as maturation of the voice-family characteristic.

stress

Initially, I had included this as uncontroversial, and feel that it's the strongest of the debatable characteristics for inclusion in an initail draft. However, its interaction with currently immature characteristics such as pitch and pitch-range, and also with speech-rate, raise the question whether this is sufficiently well-understood to include in an initial draft.

Characteristics Intentionally Dropped

cue, cue-before, cue-after

The cues can definitely be dropped in favor of generation of a series of flow objects. Whether the generated flow objects will be audio flow objects with the desired sound effect or if they will be pauses with background effects is uncertain, but either is preferable to these characteristics; the ACSS specification mentions that the :before and :after pseudo-elements would be preferable to these properties.

speak-date, speak-time

This covers the same ground as discussion on the W3C's www-html list. The date and time should be text generated by the user agent as appopriate for the language of the document; once generated, it can be read as any other text can be. Special characteristics to govern this, possibly at odds with the native language of the document, are undesirable in my opinion.

Immature Characteristics

voice-family, pitch, pitch-range, richness

According to the ACSS specification, these characteristics aren't fully understood, and so are unsuitable for inclusion in XSL right now. Full specification of speaking voice is desirable in the long run, but is too complicated for the initial draft.

speak-numeral

The implications of this characteristic is unclear from the ACSS specification. Obviously, it controls whether to read a number digit-by-digit or as text, but the specification also provides for “none”, which is the default value, which doesn't seem logical. I also think that speak-digits would be a better name, with a Boolean value; more work is needed on this characteristic.

New Units

The ACSS specification introduces three new types of units, whose acceptance should be tied to acceptance of characteristics that need them as values.

Angular Measurement

ACSS specifies three units of measurements for angles: degrees, given as “deg”, gradians, given as “grad”, and radians, given as “rad”.

Angular units are needed for these characteristics:

- azimuth
- elevation

Time

In order to specify duration, ACSS introdueces the millisecond (“ms”) and the second (“s”). They are used by

- pause-before,
- pause-after, and
- pause

but it seems, in light of other discussion, that some time measurement will be needed regardless of the exact means of specifying pauses in XSL.

Frequency

ACSS provides hertz and kilohertz (“Hz” and “kHz”) to measure frequncy. The measurements are used only by pitch, and so can probably safely be left out of the initial XSL draft.

User Agent Behavior

It is expected that a user agent with both audio and visual rendering capabilities will give the user a choice of either or both presentations. A completely blind user will not care about visuals (and may not even have his monitor on), while a sighted user in an office will probably not want any sound. However, a nearsighted user might want to have a page read, while still seeing the layout, and a driver of a car might want a quick look at pictures while the text is read.

While in a single mode, or if the user agent is incapable of visual or audio output, the user agent should at user option perform some rendering of flow objects exclusive to the other mode. For example, in visual-only mode, a user may still want to know about sound cues or background sounds, so that she may request their playing; in audio-only mode, a user may want to know about silent elements that are present. The default behavior should be to follow the stylesheet designer's instructions, though a user should be able to change the default on her system.

Whether or not pause-before and pause-after are kept as characteristics, the default audio rendering of visual objects can be specified in terms of what their defaults would be if they were kept. For instance, the sequence and literal flow objects should not have any pause before or after, by default, but the paragraph flow object should have some brief pause before, perhaps one second. Table cells, similarly, should provide some pause. If audio objects are included in the July draft, a default separation should be specified for every flow object.

Braille

Braille displays or embossers are the other primary means of access for the visually impaired. However, Braille display is computationally difficult and highly language-dependent, and should not be addressed for the July draft.

Notes

1. <URL:http://www.w3.org/TR/WD-acss-970630>

2. <URL:http://www.w3.org/TR/REC-CSS1-961217>