2018
The Journal on
Technology and
Persons with
Disabilities

Media Player Accessibility: Summary of Insights from Interviews & Focus Groups

Terrill Thompson

University of Washington

tft@uw.edu

Abstract

Researchers conducted 13 interviews and focus groups with 37 individuals, primarily at the 2017 CSUN Assistive Technology Conference, to develop a better understanding of how persons with disabilities interact with video players. Insights that emerged include the following: (1) Screen reader users need better ways to seek to a new point in the media; a text input field might offer an effective solution; (2) Screen reader users prefer human-narrated audio description over synthesized speech, but if the latter is deployed, using screen readers to read the description text is a problematic method; (3) Screen reader users need access to captions and subtitles; (4) In order for a synchronized sign language window to be effective, users should be able to control its size, position, and opacity so they can place it in the perfect position relative to the video; (5) Having too many controls adversely affects usability, but there is no agreement as to which controls are expendable. One possible approach to address this problem is to offer users the ability to add/remove controls within Preferences. A recurring theme in the discussions is that of user preferences. Individuals should be able to customize their interface to best meet their unique needs.

Keywords

Web accessibility, video accessibility, captioning, audio description, sign language, usability

Journal on Technology and Persons with Disabilities
Santiago, J. (Eds): CSUN Assistive Technology Conference
© 2018 California State University, Northridge

Introduction

Many people have difficulty fully accessing video content unless it includes specific accessibility features. The World Wide Web Consortium (W3C) Web Content Accessibility Guidelines (WCAG) 2.0 (Caldwell, et al.) includes multiple success criteria that apply to online media accessibility. For example, online video requires captions, transcripts, audio description, and synchronized sign language in order to meet the guidelines at various levels. HTML5 provides some of the necessary markup for delivering accessible video, including the <track> element, which can be used in conjunction with the <video> element to identify five distinct kinds of text tracks that can be synchronized with the video: Captions, subtitles, descriptions, chapters, and metadata (Hickson, et al.). All major browsers now support captions, and some support subtitles. However, none support all of the accessibility features made possible with HTML5. To address the gap between user needs and current support by browsers, the author of this paper created Able Player, a free open-source HTML5 media player (Able Player). Able Player has fully accessible player controls and where necessary, uses ARIA (Craig and Cooper) to expose interface elements to screen readers. Able Player is the only media player that fully supports the HTML5 <track> element, including all five kinds of text tracks (Thompson).

The goal of the current project was to develop a better understanding of how persons with disabilities interact with Able Player, and with online video players in general. The researchers conducted interviews with 37 persons with disabilities in early 2017, including 22 who are blind, 8 with low vision, 4 who are deaf or hard of hearing, 1 with physical disabilities that limit use of hands, and 2 who did not disclose their disabilities or use assistive technologies. Thirteen one-hour sessions were conducted, including 10 focus groups (with up to 6 participants each) and 3 individual interviews. One of each of these types of sessions was conducted on the University of

Washington campus; all others were conducted at the CSUN Assistive Technology Conference (California State University Center on Disabilities). In each of the sessions, participants were asked to describe their preferred technologies for accessing video online (e.g., device, operating system, browser, assistive technologies), where they tend to access video online, and their likes and dislikes about the media players they encounter on those sites. Participants had been asked to “bring your own device,” and a wide variety of devices (Windows laptops, iPads, iPhones, and Android phones) and assistive technologies (various screen readers, ZoomText, and Dragon NaturallySpeaking) were represented in the sample. Participants were asked to try one or two of three online demos that were available. Each of the demos featured a simple web page with the same six-minute video loaded into Able Player. Participants were specifically asked to “explore the video player for two to three minutes.” If they had questions during the demo, staff responded. Staff also pointed out specific features of interest (e.g., audio description, sign language, full screen, preferences) if, after two to three minutes, participants had not independently discovered these features. The specific demos participants were asked to try, and the specific features of interest that were pointed out to participants, depended on participants' individual characteristics: Demo 1 featured a video that included audio description, provided as a separate human-narrated video; Demo 2 included text-based audio description to be read aloud by users' screen readers; Demo 3 featured closed captions and synchronized sign language. After exploring the demo(s), participants were asked to describe their experience and talk about features they found most and least appealing, and improvements they would recommend.

Discussions

Key ideas that emerged from participant discussions tended to relate to one of six issues. Ideas are summarized below, grouped by issue in no particular order.

Issue #1: Seeking to a new point in the media

Several screen reader users said they experience frustration using the seek bar with most media players. Seek bars typically work by moving a slider left or right along a horizontal bar. Often this requires a mouse, or if it's keyboard-accessible, it provides insufficient feedback to be usable by screen reader users. The developers of Able Player have addressed this need in two ways. First, the seek bar is coded using applicable ARIA markup such as role="slider" and related attributes (King, et al.). Second, the seek bar is one of two ways to seek, the other being a pair of buttons labeled "Rewind" and "Forward." By default, both buttons and the seek bar move in increments equal to a percentage of the video duration, calculated automatically to balance precision with practicality (e.g., short videos have fewer stops than longer videos). After testing both seek methods, users generally rated these features favorably. However, many felt the interface could be improved in two ways.

First, the player currently fails to disclose the seek interval so users don't know where they will end up if they click a button or move the slider. This could be addressed by simply specifying the seek interval in the label or tooltip (e.g., "Forward 10 seconds"). Another option would be to offer preferences that allow users to define the jump unit (e.g., "seconds", "percent of video duration") as well as the value. For example, if users select a jump unit of "seconds" and value of "10," the seek interval would be 10 seconds.

Second, jumping in intervals, regardless of how the interval is defined, lacks the granularity that users sometimes need. If they need to jump to a very specific point in the video (defined in seconds or microseconds), this will be very difficult or impossible with an interval-based interface. Discussions yielded some possible alternative interfaces. One suggested by several users was a “Jump to time” edit box that would allow users to enter a specific time value in an intuitive format such as *hours:minutes:seconds*.

Furthermore, the participant who uses Dragon NaturallySpeaking expressed interest in using natural language commands like “Forward to eight minutes” or “Rewind to five thirty-seven.” Whether this sort of functionality should be handled by the media player or the speech-to-text tool requires further consideration.

Issue #2: Audio Description Preferences

Able Player includes a “Description” button that enables users to turn audio description on or off. This works differently depending on whether the description is provided as an alternative described version of the video (Method A) or as a timed text track (Method B). With Method A, the user receives the described version if the Description button is toggled on; otherwise they receive the non-described version. With Method B, the state of the Description button determines whether the description text is exposed to screen reader users.

Screen reader users who participated in the study said they were intrigued by the idea of having audio description read aloud by their screen readers; however, they noted several fundamental problems with this idea. One participant said they had learned to tune out their screen reader when it starts talking unexpectedly, as this often indicates an incoming message or other alert that really isn't critical. If this happens while watching a video, they suspected they would just ignore it and might take action to silence it. Another participant argued that the role of screen readers is to respond to user's commands, and while there can be latitude for alerts and other critical information, automatic reading of content generally falls outside this scope. This same participant demonstrated that pressing a key while description is being read interrupts the description, yet the video continues to play (without description, thereby breaking the accessibility feature). This illustrates why screen reader functionality and audio description playback should be separate.

Text-based description may still be viable, as producing it is much easier and cheaper than recording a human narrator to voice the description content. However, it may be preferable for the description to be self-voiced (e.g., with synthesized speech built into the media player, browser, or operating system) rather than read by the user's screen reader. Another benefit of self-voicing is that it can more gracefully support extended audio description, a technique in which the video is paused temporarily while description is read. Extended audio description is necessary if a video has too little suitable time to inject audio description content. Currently, Able Player handles this need by automatically pausing the video when description starts (a feature that can be enabled within Preferences). However, since Able Player depends on screen readers to read the description, it can't automatically resume playback because there is no mechanism by which screen readers can communicate when they're finished reading. Therefore, users have to resume playback themselves by pressing the spacebar. If a self-voicing solution were to be used, the duration of the voicing could be known and playback could resume automatically.

When screen reader users were asked whether, if given a choice, they would prefer human-narrated description or text-based description, all but one of the participants said they prefer human-narrated description. When asked why, they explained that human narrators speak with more natural inflection and emotion and are more likely to pronounce words accurately, whereas synthesized speech is more likely to be distracting. If asked whether this preference applied equally to dramatic and instructional content, all participants agreed that it was more applicable to dramatic content. Also, many participants were quick to stress that description by any method is better than no description at all.

One user reported a preference for text-based description because they want to be able to review it. They described a hypothetical interface in which the user is watching a video that includes text-based description. The user hears a block of detailed described content and wants to go back to review it, presses a keystroke that triggers a “Review description” function, which pauses the video and places focus at the start of the description block, and resumes playback by pressing the spacebar when finished.

Issue #3: Accessibility of Subtitles to Screen Reader Users

Although Able Player exposes text-based audio description to screen readers in sync with the video, it does not expose caption or subtitle text to screen readers. Currently, Able Player hides this content from screen reader users (using aria-hidden=“true”), based on the belief that a screen reader reading content in synch with spoken word in the program audio would be distracting and undesirable. Instead, Able Player reassembles all timed text content (chapters, descriptions, and captions/subtitles) into an interactive transcript, which can be read independently of the media playback.

However, one participant argued for captions and subtitles to be accessible to screen reader users, as these features provide critical content and functionality. Foreign language subtitles are particularly critical if the audio is in a language the user doesn't understand. Same-language captions can be similarly helpful, for example if the spoken audio is difficult to hear. If the various audio tracks have volume that can be controlled independently, a user could reasonably watch a video with the audio turned down slightly in favor of a louder caption or subtitle track, self-voiced with speech synthesis.

Issue #4: Synchronized Sign Language

Able Player currently supports sign language in a separate window. The window contains a separate video (e.g., a filmed video of a person translating the video into sign language). Both videos are operated by the same controller so they remain synchronized during playback. The sign language window can be toggled on/off using a “Sign Language” button on the player control bar. The window can be dragged or resized using either mouse or keyboard.

Currently the sign language video is fully opaque. Therefore, if users drag and resize the sign language window so that it's positioned on top of the program video, it obstructs any content that might appear behind it. Participants who are deaf or hard of hearing recommended including a user preference for sign language window transparency. If the signer's hands have sufficient contrast with their clothing, the entire window can be made semi-transparent without compromising understandability.

Participants suggested offering even greater customizability of the sign language window. In addition to having the ability to move and resize the window, users should be able to crop it in order to reduce the size of the video to the smallest size possible without compromising understandability. Users should also be able to resize the sign language window *without* preserving the aspect ratio.

Participants also suggested making it possible to move and resize the video window, as is currently possible with the sign language window. For sign language users, the optimum position of the video and sign language windows relative to each other may depend on being able to reposition both objects.

Issue #5: Quantity/Visibility/Order of Controls

One consequence of a media player having a large quantity of accessibility features is that it has a corresponding large quantity of buttons and controls, which can impede usability. One recurring suggestion was to hide less frequently used buttons behind a “More” button, exposing them to users upon request. However, when participants who suggested this approach were asked which buttons should be hidden, there was no consensus. Even buttons that are seldom used by individuals without disabilities (e.g., the Description and Sign Language buttons) were not considered “second-tier” by users who depend on those features. This suggests that there is no “one size fits all” arrangement. One participant suggested applying some intelligence to the problem and showing only the user's most frequently used buttons by default. However, this would result in a non-standard interface that varies across users, potentially changes over time, or perhaps worse, is self-fulfilling as hidden icons are less likely to be used, even if they might be useful. Another recurring suggestion was to provide users with the ability to control which buttons are included (e.g., a set of checkboxes within the Preferences dialog for all buttons and controls); users could toggle on and off which features they don't need or want.

Participants with eyesight differed on whether they wanted to see buttons on the screen all the time. Individuals with low vision expressed a preference for visible controls, as it can be difficult for them to find things that appear and disappear. Individuals with no reported visual impairments tended to prefer controls that are hidden during playback, especially in full screen mode. One participant felt very strongly about this, as they find the controls to be distracting and a barrier to their getting fully immersed in the video. All participants who favored hiding controls during playback felt that they should be easily recoverable (e.g., by pressing a key or moving the mouse).

Conclusion

A recurring theme in each of the discussions is that of user preferences. Individual users have unique needs. They should therefore be able to customize their interface and experience to best meet their needs.

Acknowledgments

This chapter is based on work supported by the National Science Foundation (grant number CNS-1539179). Any opinions, findings, and conclusions or recommendations are those of the author and do not necessarily reflect the policy or views of the federal government, and you should not assume its endorsement.

Works Cited

Able Player. “Able Player.” *Able Player*, n.d., http://ableplayer.github.io/ableplayer/. Accessed 30 Nov. 2017.

Caldwell, Ben, et al., editors. “Web Content Accessibility Guidelines 2.0.” *W3C*, 2008, www.w3.org/TR/WCAG20/. Accessed 30 Nov. 2017.

California State University Center on Disabilities. “CSUN Assistive Technology Conference.” *CSUN*, 2017, www.csun.edu/cod/conference/2017/sessions/. Accessed 30 Nov. 2017.

Craig, James and Cooper, Michael, editors.” Accessible Rich Internet Applications (WAI-ARIA) 1.0.” *W3C*, 2014, www.w3.org/TR/2014/REC-wai-aria-20140320/. Accessed 30 Nov. 2017.

Hickson, Ian, et al., editors. “HTML5.” *W3C*, 2014, www.w3.org/TR/HTML5/. Accessed 30 Nov. 2017.

King, Matt, et al., editors. “Slider.” *WAI-ARIA Authoring Practices 1.1*, W3C Working Draft 28 June 2017, www.w3.org/TR/wai-aria-practices-1.1/#slider/. Accessed 30 Nov. 2017.

Thompson, Terrill. “Video for All: Accessibility of Video Content and Universal Design of a Media Player.” In S. Burgstahler (Ed.), *Universal design in higher education: From principles to practice, second edition*, 2015, Boston: Harvard Education Press, pp. 259-273.

© 2018 California State University, Northridge