

# *State of Media Accessibility in HTML5*

Silvia Pfeiffer

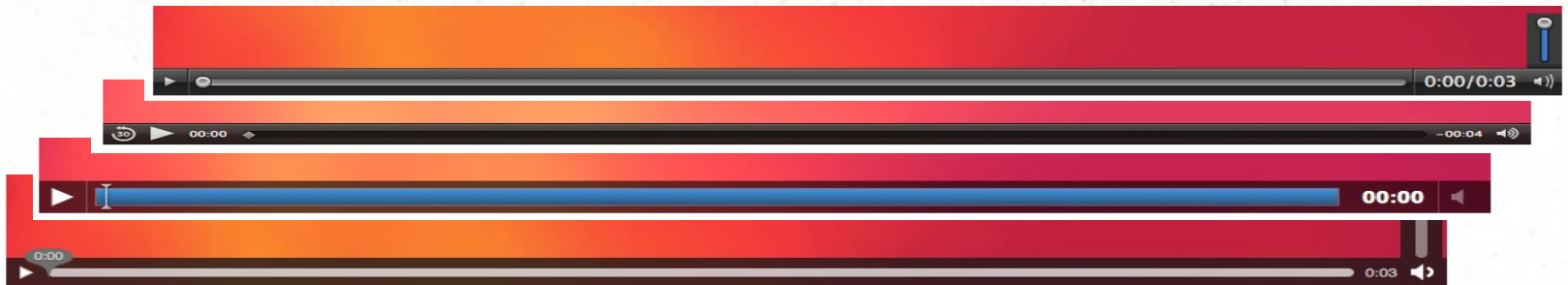
OVC 2010

Annodex, Xiph, Vquence, Mozilla

# *The HTML5 <video> tag*

Example:

```
<video poster="video.png" controls>  
<source src="video.ogv" type="video/ogg">  
<source src="video.mp4" type="video/mp4">  
</video>
```



# *Accessibility Aspects*

A ... Vision-impaired users

B ... Hard-of-hearing users

C ... Deaf-blind users

D ... Learning Disabilities

*A ...*



*Vision-  
impaired  
users*

# *Vision-impaired users*

## Described video

“descriptive narration of key  
visual elements”

# *(Extended) Audio description*



<http://web.mac.com/eric.carlson/w3c/NCAM/extended-audio.html>

# *Open Audio Description*

```
<video poster="video.png" controls>  
<source src="video.ogv" type="video/ogg">  
<source src="video.mp4" type="video/mp4">  
</video>
```

# *In-band Audio Description*

Multitrack with the original content:

```
<video poster="video.png" controls>  
<source src="video.ogv" type="video/ogg">  
<source src="video.mp4" type="video/mp4">  
</video>
```

Needs JS such as:

```
if (video.audiotracks[1].kind == "descriptions")  
desc = video.audiotracks[1]; // => AudioTrack
```



# *External Audio Description*

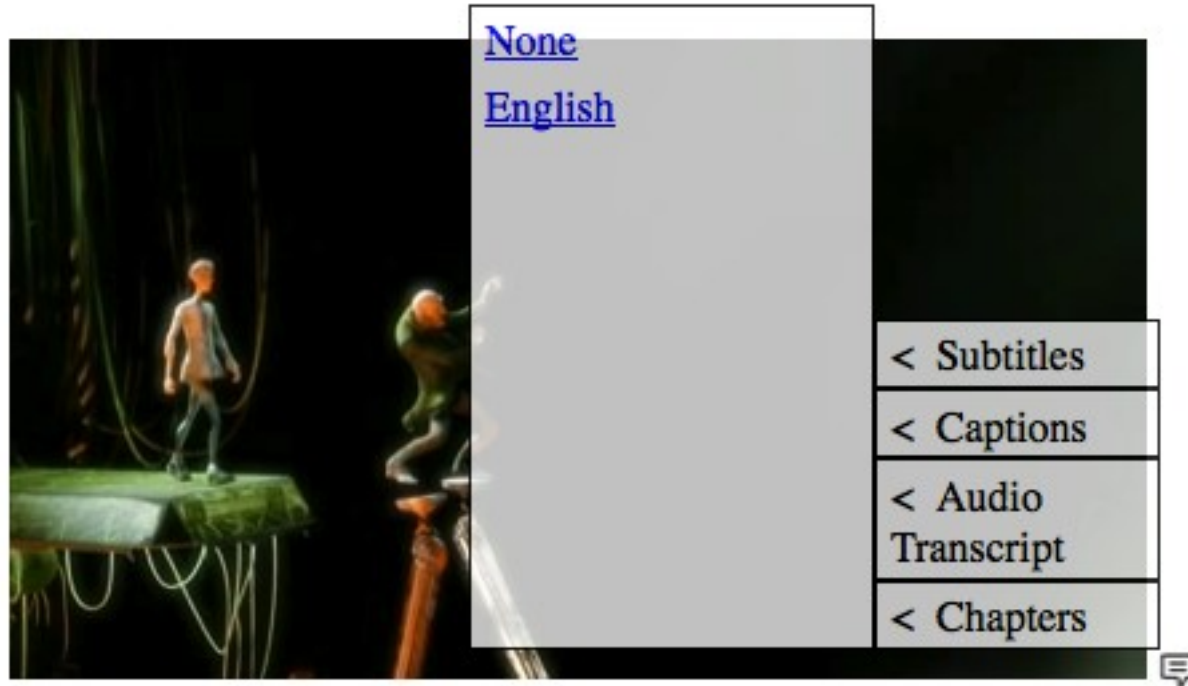
Needs markup such as:

```
<video poster="video.png" controls>  
<source src="video.ogv" type="video/ogg">  
<source src="video.mp4" type="video/mp4">  
<audiotrack kind="descriptions" srclang="en">  
<source src="description.ogg" type="audio/ogg">  
<source src="description.mp3" type="audio/mp3">  
</audiotrack>  
</video>
```

Combined with JS such as:

```
audiotrack =  
getElementsByTagName("audiotrack")[0].track;  
audiotrack.mode = SHOWING;  
alert(audiotrack.kind);      // => "descriptions"  
alert(audiotrack.language); // => "en"
```

# *(Extended) Text description*



[http://www.annodex.net/~silvia/itext/elephant\\_no\\_skin\\_v2.html](http://www.annodex.net/~silvia/itext/elephant_no_skin_v2.html)

# *In-band Text Description*

Multitrack with the original content:

```
<video poster="video.png" controls>  
<source src="video.ogv" type="video/ogg">  
<source src="video.mp4" type="video/mp4">  
</video>
```

Specified markup:

```
if (video.tracks[1].kind == "descriptions")  
description = video.tracks[1]; // => TimedTrack
```

# *External Text Description*

Specified extra Markup:

```
<video poster="video.png" controls>  
<source src="video.ogv" type="video/ogg">  
<source src="video.mp4" type="video/mp4">  
<track kind="descriptions" srclang="en"  
src="descriptions.wsr" label="English  
Descriptions">  
</video>
```

Select specified new JavaScript:

```
track = getElementsByTagName("track")  
[0].track;  
track.mode = SHOWING;  
alert(track.kind);      // => "descriptions"  
alert(track.language); // => "en"
```

# File format: WebSRT

00:00:00.00 --> 00:00:05.00

The orange open movie project presents

00:00:05.01 --> 00:00:12.00

Introductory titles are showing on the background of a water pool with fishes swimming and mechanical objects lying on a stone floor.

00:00:11.01 --> 00:00:14.80

elephants dream

# *Vision-impaired users*

## Navigation and Interaction





## **activate / deactivate descriptions**

1. `description.mode = HIDDEN / SHOWING;`
2. `@controls` should include features to “turn on audio descriptions”

## **navigate within and into the content**

“a fragment identifier can be used to indicate a start position”

Specified media fragment URIs:

<http://example.com/video.ogv#t=40>

keyboard control

**navigate between alternative content**

multiple tracks at different resolution

**@controls** allow navigation between them

**navigate out of content**

hyperlinks in cues



*B ...*

*Hard-of-hearing  
users*

# *Hard-of-hearing users*

## Captions

“time-synchronized text cues”

# *(Enhanced) Captions*



[http://www.youtube.com/watch?  
v=dtyXYjXNI2w](http://www.youtube.com/watch?v=dtyXYjXNI2w)

# *Open Captions*

```
<video poster="video.png" controls>  
<source src="video.ogv" type="video/ogg">  
<source src="video.mp4" type="video/mp4">  
</video>
```



# *In-band Captions*

Multitrack with the original content:

```
<video poster="video.png" controls>  
<source src="video.ogv" type="video/ogg">  
<source src="video.mp4" type="video/mp4">  
</video>
```

Specified new JS:

```
if (video.tracks[1].kind == "captions")  
caption = video.tracks[1]; // => TimedTrack
```

# *External Captions*

Specified new markup:

```
<video poster="video.png" controls>  
<source src="video.ogv" type="video/ogg">  
<source src="video.mp4" type="video/mp4">  
<track kind="captions" srclang="en"  
src="captions.wsrt">  
</video>
```

Specified new JavaScript:

```
track = getElementsByTagName("track")  
[0].track;  
track.mode = SHOWING;  
alert(track.kind);      // => "captions"  
alert(track.language); // => "en"
```

# Proposed file format: WebSRT

00:00:15,000 --> 00:00:17,951

At the left we can see...

00:00:18,166 --> 00:00:20,083

At the right we can see the...

00:00:20,119 --> 00:00:21,962

...the head-snarlers

00:00:21,999 --> 00:00:24,368

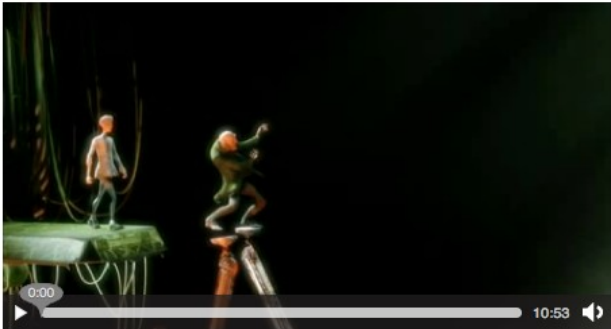
Everything is safe.

Perfectly safe.

*Hard-of-hearing users*

Transcripts

# *Linked Transcripts*



Click on text to play video from there.

[Screen text: "The orange open movie project presents"]

[Introductory titles are showing on the background of a water pool with fishes swimming and mechanical objects lying on a stone floor.]

[Screen text: "Elephant's Dream"]

Proog: At the left we can see... At the right we can see the... the head-snarlers. Everything is safe. Perfectly safe. Emo? Emo!

Proog: Watch out!

[The old man, Proog, shoves the younger and less experienced Emo on the ground to save him from being mowed down by a barrage of jack plugs that whirl back and forth between the two massive switch-board-like walls.]

[http://localhost/~silvia/mozilla/a11y\\_best\\_practice/demo2\\_transcript.html](http://localhost/~silvia/mozilla/a11y_best_practice/demo2_transcript.html)

# *Linked Transcripts*

```
<video poster="video.png" controls aria-  
describedby="transcript" src="video.ogv">
```

```
</video>
```

```
<div id="transcript" style="overflow-y:scroll;">
```

```
<p id="c1" class="cue" data-time="15.0">Proog:
```

```
At the left we can see... At the right we can see  
the... the head-snarlers. Everything is safe.
```

```
Perfectly safe. Emo? Emo!</p>
```

```
</div>
```

Navigate video from transcript:

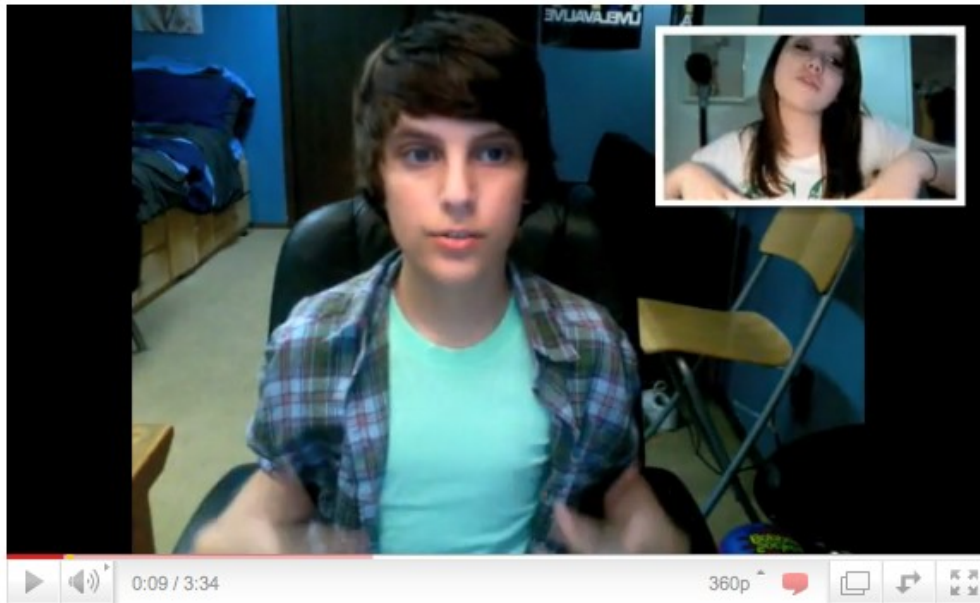
```
cues[i].addEventListener("click",  
function(evt) {  
var start =  
parseFloat(this.getAttribute("data-time"));  
video.currentTime = start;  
video.play();  
}, false);
```



## Navigate transcript from video:

```
video.addEventListener("timeupdate",
function(evt) {
  for (i=0; i<cues.length; i++) {
    if (video.currentTime >=
parseFloat(cues[i].getAttribute("data-time")) &&
        video.currentTime <
parseFloat(cues[i+1].getAttribute("data-time")))
    {
      // scroll cues[i] to center of display
    }
  }
}, false);
```

# *Hard-of-hearing users*



Sign  
translation  
video

<http://www.youtube.com/watch?v=IZjBqVn4AQL>

# *In-band Sign Language*

Multitrack with the original content:

```
<video poster="video.png" controls>  
<source src="video.ogv" type="video/ogg">  
<source src="video.mp4" type="video/mp4">  
</video>
```

Needs new JS such as:

```
if (video.videotracks[1].kind == "signing")  
caption = video.videotracks[1]; // => VideoTrack
```

# *External Sign Language*

Needs new markup such as:

```
<video poster="video.png" controls>  
<source src="video.ogv" type="video/ogg">  
<source src="video.mp4" type="video/mp4">  
<videotrack kind="signing" srclang="asl">  
<source src="signing.ogv" type="video/ogg">  
<source src="signing.mp4" type="video/mp4">  
</videotrack>  
</video>
```

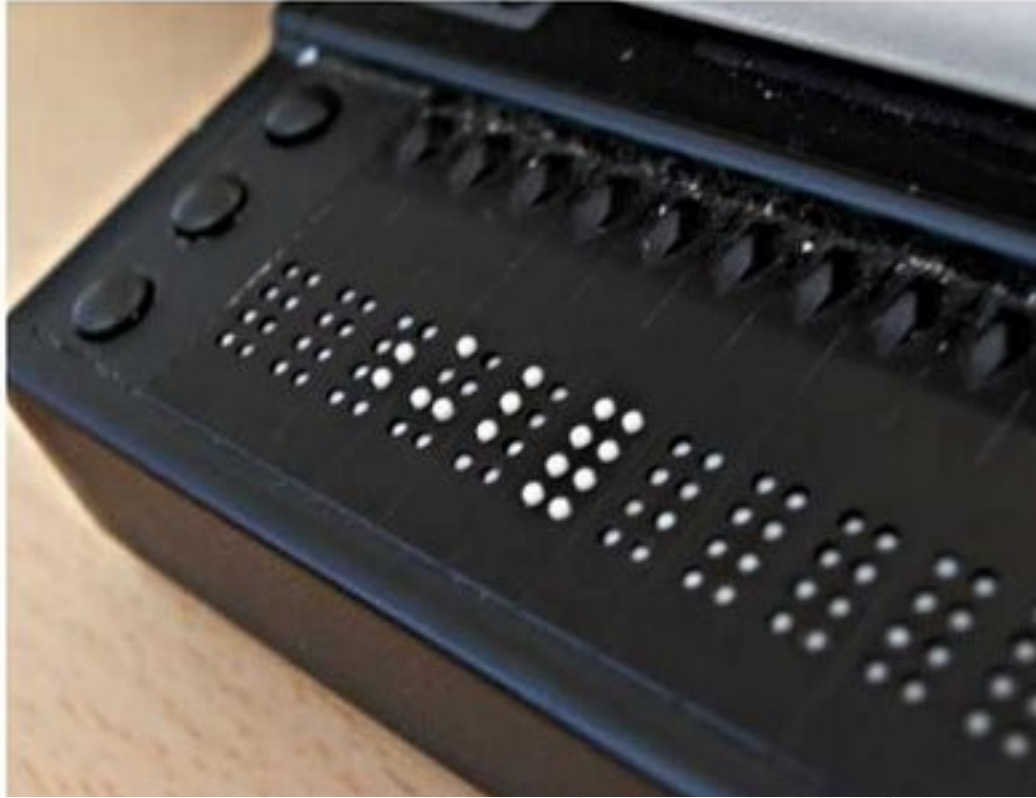
Combined with new JavaScript, e.g.:

```
track = getElementByTagName("videotrack")  
[0].track;  
videotrack.mode = SHOWING;  
alert(videotrack.kind);      // => "signing"  
alert(videotrack.language); // => "asl"
```

*C ...*



*Deaf-blind  
users*



Transcription with text description and caption information rendered to braille

*D ...*



*Learning  
Disabilities and  
other needs*



# *Special Needs*



Clear audio  
“amplify speech separately”

# *Special Needs*



Time-scale  
modification

“speed up/slow down  
audio without  
changing pitch”

# *Special Needs*

## Interactive card trick



Hey, a secret link! cool!

I need six cards for this trick.

This is an interactive video! so DONT sit back... INTERACT!

0:10 / 1:10

Rate: ★★★★★  
11 ratings

Views: 788  
this video has [annotations](#)

The image shows a video player interface. The video content features a man in a grey shirt performing a card trick. There are two text annotations overlaid on the video: one at the top right that says "Hey, a secret link! cool!" with a small red icon, and another in the center that says "I need six cards for this trick." Below the video, there is a progress bar showing 0:10 / 1:10. At the bottom, there is a rating section with five stars and the text "Rate: ★★★★★ 11 ratings" and a views section with the text "Views: 788 this video has annotations".

Attaching  
Explanations  
“learning  
through  
additional  
material”

# *Status in HTML5*

<b>Feature</b>	<b>Markup</b>	<b>JavaScript</b>	<b>WebSRT</b>	<b>User Preferences</b>	<b>User Interface</b>
(ext) audio description OPEN	OK				
(ext) audio description IN-BAND		TODO		recommended	
(ext) audio description EXTERNAL	TODO	TODO		recommended	
(ext) text description IN-BAND		TimedTrack		recommended	
(ext) text description EXTERNAL	<track>	TimedTrack	Yes	recommended	
activate/deactivate video descriptions	@controls	mode= SHOWING/ HIDDEN		recommended	
navigate into and within content	media fragment URI	video.currentTime			insufficient
navigate between tracks	insufficient	mode			insufficient
navigate out of content			insufficient		

# *Status in HTML5*

<b>Feature</b>	<b>Markup</b>	<b>JavaScript</b>	<b>WebSRT</b>	<b>User Preferences</b>	<b>User Interface</b>
Open captions	OK				
In-band (enhanced) captions		TimedTrack		recommended	
External (enhanced) captions	<track>	TimedTrack		recommended	
plain transcript	OK				
linked transcript	Aria-describedby, data-time	currentTime, data-time	possible		
In-band sign language		TODO		recommended	
external sign language	TODO	TODO		recommended	
slow down content	possible	possible			possible
In-band clear audio		possible		possible	possible
External clear audio	possible	possible		possible	possible