TTML Animation

G. Adams

TPAC2013

TTML1 Animation (1)

- attribute target/value shorthand
  - SVG separates specification of target attribute name and value, e.g.,
    <set attributeName="tts:color" to="red"/>
  - TTML1 uses style property expression shorthand:
    <set tts:color="red"/>
- multiple attribute targets
  - SVG permitted only one attribute target per animation element
  - TTML1 introduces multiple attribute targets per animation element, e.g.
    <set tts:opacity="0.3" tts:origin="30% 40%"/>

TTML1 Animation (2)

- single element target per animation element
  - element target is parent element, e.g. <p><set/></p> targets parent <p/>
- inline animation

<p tts:visibility=“hidden”>
<set begin=“2s” tts:visibility=“visible”/>
Text starts hidden, becomes visible 2s later.
</p>

TTML1 Animation (3)

- discrete animation

  <set begin="1s" tts:origin="100px 100px"/>
  <set begin="2s" tts:origin="200px 200px"/>

  – step value function only

TTML2 Animation

- Backward compatible with TTML1.
- Introduces
  - indirect target attribute
  - explicit target element
    - out-of-line declarations
    - grouping element for out-of-line declarations
  - continuous (smooth) animation
- Restrict functionality to enable mapping to CSS Animations

Indirect Target Attributes (1)

- ISSUE 165 – See CP1
  - http://www.w3.org/wiki/TTML/changeProposal001

instead of

<p>

<set tts:opacity="0.3"/>

<set tts:color="red"/>

</p>

Indirect Target Attributes (2)

use

<head>

<style id=“s” tts:opacity=“0.3” tts:color=“red”/>

</head>

…

<p>

<set style=“s”/>

</p>

Explicit Target Element (1)

- Explicit target element specification

```
<set href="div2" .../>
```

- Where to specify?

```
<head>
  <animation>
    <set href="div2" .../>
  </animation>
</head>
```

- Referred to as *out-of-line* animation.

Explicit Target Element (2)

- Need for explicit target in inline context? e.g.

instead of

```
<p>
  <span id="span1"><set .../>ABC</span>
  <span id="span2"><set .../>XYZ</span>
</p>
```

Explicit Target Element (3)

use

<p>
<set href="span1" .../>
<set href="span2" .../>
ABC
XYZ
</p>

Explicit Target Element (4)

- Should <animation/> be permitted elsewhere, i.e., outside of <head/>?
- Raises higher level questions:
  - Should <styling/> and <layout/> be permitted elsewhere?
  - How about multiple <head/> and <body/>, e.g.,

```
<tt>
  <head/>
  <body/>
  <head/>
  <body/>
</tt>
```

Explicit Target Element (5)

- More advanced targeting?, e.g.,

  Web Animations introduces MQ selection:
  <set select="p"/>
  <set select="p.class"/>

- If so, then probably should introduce @class.

Continuous Animation (1)

- Multiple issues drive requirement for continuous (non-discrete) animation:
  - Issues 22, 23, 72, 227
- TTML1 adopts SVG discrete animation syntax, so it is reasonable to adopt continuous animation syntax from SVG as well.
- However, need to constrain semantics to CSS Animation capabilities. Can also simplify by not supporting shorthands.

Continuous Animation (2)

- The SVG <animate/> element defines certain functionality not readily translatable into CSS Animation:
  - @accumulate – cumulative on repeat
    - none | sum
  - @additive – add base and multiple
    - replace | sum
- Propose to defer this functionality until supported by CSS.

Continuous Animation (3)

- SVG supports five value specification syntaxes:
  - @values
  - @from, @to
  - @from, @by
  - @by
  - @to
- The last four are shorthand expressions for subsets of @values.
- Propose that only @values be supported.

Continuous Animation (4)

- SVG supports two repetition specifications:
  - @repeatCount
    - Real-valued repetition count or **indefinite**. Fractional part denotes portion of simple animation duration.
  - @repeatDur
    - Real-valued total duration.
- Propose supporting only @repeatCount, since @repeatDur provides no additional semantics.

Continuous Animation (5)

- Need to introduce multi-valued style property expressions, e.g.,

  <animate tts:color="red; green; blue"/>

  <style id="s" tts:origin="0% 0%; 25% 25%"/>
  …
  <animate style="s"/>

Continuous Animation (6)

- @calcMode="paced"
  - Can be expressed as @calcMode="linear" with explicit @keyTimes that represent equivalent to paced mode.