Integrating Timing into XML Documents

Patrick Schmitz
MS Research
BARC Telepresence

BARC Telepresence

- Traditional Telepresence
  - Reliable Multicast
  - Gaze-corrected videoconferencing
- New Directions
  - Next generation Media services
  - Multi-modal documents
  - Representing *time* in documents

Timing Integration

- Motivation for a common model
- Requirements
- Background – SMIL evolution
- Example: HTML+SMIL
- Approaches to Integration
- Future work and applications

Motivation

- Common authoring semantics
- Leveraging SMIL in HTML/CSS content
- Providing Time model in documents
  - One clock for the whole document
- Synchronizing HTML to TV

Requirements

- Need common, straight-forward authoring models
- Need flexible approach to syntax
  - Sometimes want inline syntax
  - Sometimes want to model like style
  - Sometimes need a separate document
- Need to be able to mix approaches

Background

- SMIL 1 – a first step
  - Simple timing model and syntax
  - Standalone, no integration support
- HTML+TIME submission
  - Proposal to extend SMIL and integrate with HTML
- SMIL Boston
  - New support for interaction, sync control
  - Modularization provides means of sharing critical pieces among languages
  - Includes HTML+SMIL Language

Integration example: HTML+SMIL

- HTML+SMIL allows Web developers to:
  - Control DHTML properties along a timeline
  - Integrate media as part of their page description
  - Synchronize media elements and actions in the page
- Reduces dependency on scripting as a way of controlling animation of properties

Demos

- AOE Banner ad
- Expanding ad images
- Business presentations

HTML+SMIL Timing and Media Markup

- Media elements – `video`, `audio`, et al.
- Timing elements – `par`, `seq`, `excl`
  - `timeContainer` attribute
- Timing attributes – `begin`, `end`, et al.
  - Applied to most HTML content
  - Includes event-based/interactive declaration
- Animation support
  - CSS properties, motion, effects, etc.

What does `begin` mean for `div` or `strong`?

- `timeAction` controls semantics of adding timing to HTML elements
  - `intrinsic` – defined for phrasal and presentation elements, reverts to visibility for text, div, media, etc.
  - `display`, `visibility` control style
  - `style` controls inline style (CSS/XSL)
  - `class` adds class name to class set

Approaches to Integration

- Inline syntax
  - Attributes added to language elements
  - Approach used in SMIL, HTML+SMIL
- Styled Timing
  - CSS or XSL stylesheets used to apply timing to a language
- Timesheets
  - Separate timing from both content and presentation style

Inline Syntax approach

- Easy to understand, easy to author
- Generalized, extensible semantics with `timeAction`
- Better when document structure aligned to timing structure
- Can also be used to override or augment styled timing
  - Requires aligned model of timing properties

Styled Timing approach

- Useful when document structure aligns closely with timing structure
  - Example: sequence of highlights on list
- CSS 3 required to manipulate timing properties
  - CSS3 may include SMIL module
  - XSL FO applies much the same
- General need: filter chain model
  - Universal cascade and view model

Issues with Styled Timing

- Must preclude feedback loops
  - If timing controls style, and style redefines timing, what should happen?
  - Possible solution: lock timing properties when applying timeAction
- Specifying the side-effects
  - When does the effect of timing ripple through CSS, XSL, etc.???
  - General problem common to animation

Timesheets approach

- Abstract timing away from content and presentation style
  - SMIL timing, `timeAction`, *no media*
- Useful when:
  - document structure and timing structure do not align
  - synchronization spans multiple documents
  - inline syntax impractical or illegal
    - Copyright restrictions
    - Digital Talking Books

Issue: Interpreting multiple references to an element

- Multiple orthogonal `timeActions` are easy – just do all of them
- Multiple instances problematic
  - Synthesize elements? (*No!*)
- Conclusion: model as animation
  - Requires property-based model for `timeAction`
  - Simplified animation "sandwich"
    - Use activation priority, no composition

Other Issues

- How to combine Timesheets with other approaches?
  - Inline and Styled Timing define a *cascade*
  - Inline markup overrules Styled Timing
  - Timesheets define an *additive* model
    - Do we need composition tools to combine with other approaches?
- Defining sync among documents
  - ITV model of HTML synced to TV
  - Digital Talking Books

Conclusions

- Common model based upon `timeAction` is essential
- Cascade rules combine Styled and Inline Timing approaches
  - Models timing as properties a la CSS
- Timesheets layer, rather than override
  - Timesheets can specify multiple actions, using animation semantics
- W3C *must* define filter chain

Future work & Applications

- Resolve Issues and write the spec!
  - WWW9 Workshop looking at this.
- Potential customers
  - DTB, eBooks, multimodal documents
- Named Timespaces
  - Defines sync and interrupt semantics
  - Synchronizes HTML+SMIL to broadcast television, DVD and CD content
  - Tool for accessible multimedia?

Resources

- Specs available at http://www.w3.org/TR/smil-boston
- Early implementation in IE 5.5 (HTML+TIME)
  - IE docs/demos at msdn.microsoft.com
- SMIL Animation in SVG viewers
- WWW9 workshop on multimedia
  - http://www.cwi.nl/~lynda/www9/