SMIL Timesheets 1.0

W3C Working Group Note 28 March 2012

This version:
http://www.w3.org/TR/2012/NOTE-timesheets-20120328/
Latest version:
http://www.w3.org/TR/timesheets/
Previous version:
http://www.w3.org/TR/2008/WD-timesheets-20080110/
Editors:
Petri Vuorimaa, TKK
Dick Bulterman, CWI
Pablo Cesar, CWI.

Abstract

This document defines an XML timing language that makes SMIL 3.0 element and attribute timing control available to a wide range of other XML languages. This language allows SMIL timing to be integrated into a wide variety of a-temporal languages, even when several such languages are combined in a compound document. Because of its similarity with external style and positioning descriptions in the Cascading Style Sheet (CSS) language, this functionality has been termed *SMIL Timesheets*.

Status of this document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at http://www.w3.org/TR/.

Publication as a Working Group Note does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

This document has been produced by the SYMM Working Group as part of the W3C Synchronized Multimedia Activity. The goals of the SYMM Working Group are discussed in the SYMM Working Group Charter. The authors of this document are the SYMM

Working Group members. Different parts of the document have different editors.

This document was part of the SMIL 3.0 specification as the "SMIL 3.0 External timing" module, extending SMIL timing. It was removed from the SMIL 3.0 specification at Candidate Recommandation phase in order to give it more visibility as Timesheets allows integration of timing into a wide range of other XML languages. This document did not collect enough implementation experience to advance further than Last Call Working Draft. It was published as a Working Group Note at closure of the SYMM Working Group. This document represents the state of SMIL Timesheets 1.0 as of 2012. The SYMM Working Group does not expect any further changes to this document.

Comments or report errors on this document can be sent to the public mailing www-smil@w3.org - (public archives) including the prefix `'[Timesheets NOTE comment]'` in the subject line.

This document was produced by a group operating under the 5 February 2004 W3C Patent Policy. W3C maintains a public list of any patent disclosures made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains Essential Claim(s) must disclose the information in accordance with section 6 of the W3C Patent Policy.

Table of contents

1 Introduction

This section is informative

This document defines an XML-based timing language that makes SMIL 3.0 element and attribute timing control available to a wide range of other XML-based languages. This language allows SMIL timing to be integrated into a wide variety of a-temporal languages, even when several such languages are combined in a compound document. Because of its similarity with external style and positioning descriptions in the Cascading Style Sheet (CSS) language, this functionality has been termed *SMIL Timesheets*.

SMIL Timesheets can be seen as a temporal counterpart of CSS. Whereas CSS defines the spatial layout of the document and formatting of the elements,SMIL Timesheets specify which elements are active at a certain moment and what their temporal scope is within a document. And as with CSS, SMIL Timesheets can be reused in multiple documents, which can provide a common temporal framework for multimedia presentations with different contents but identical storylines. The document can be shown in a user agent even if SMIL Timesheets are not supported, since the contents and the layout are still governed by the document itself. Of course, the temporal aspect of the document is then lost, since all the elements are active all the time.

1.1 Motivation

SMIL Timesheets allows the definition of out-of-line timing in conjunction with non-SMIL languages including compound XML documents. To make authoring easier, it contains only a limited subset of SMIL functionality.

SMIL Timesheets provides a temporal dimension to existing a-temporal web documents, in the form of a external XML document. It can be used, for example, for online slide shows or photo galleries. The following [TimesheetsTutorial] provides a number of examples together with an implementation of SMIL Timesheets.

An alternative to Timesheets is XHTML+SMIL [XHTMLplusSMIL] which gives full SMIL functionality as in-line in non-SMIL XML documents. On the other hand, SMIL itself gives full SMIL functionality as in-line XML-based document.

1.2 Design Rationale

This section explains the design rationale behind Timesheets, its intended use, and its relation to other XML-based languages.

XML-based compound documents need to define the structure of the document and its content, the styling and layout of the document, and the user interaction and internal logic. Where the structure and contents are defined by the host language (XHTML) [XHTML], the styling and layout by Stylesheets (CSS) [CSS2], vector graphics support by SVG [SVG], and the user interaction and data model by XForms [XForms]. Timeshets is a solution for providing timing control capabilities for such compound

documents.

Topic	*Technology*
Structure	**Host Language (XHTML)** [XHTML]
Layout and Styling	**Stylesheets (CSS)** [CSS2]
Vector Graphics	**SVG** [SVG]
User Interaction and data model	**XForms** [XForms]
Timing	**SMIL Timesheets**

Because the intention of Timesheets is to provide timing control capabilities, they are based on the SMIL 3.0 Timing and Synchronization modules with some restrictions and additional attributes.

In addition to the SMIL 3.0 Timing and Synchronization modules, SMIL Timesheets includes the SMIL 3.0 PrefetchControl module and the SMIL 3.0 BasicAnimation module is used for controlling animations.

Moreover, SMIL Timesheets defines two new elements, not included in the SMIL 3.0 Timing and Synchronization modules:

- **timesheet**
- **item**

Finally, similary to CSS, SMIL Timesheets uses CSS selectors [CSS2] for identifying the elements in the web document to which apply timing control.

1.3 Structure of the Document

Firstly, an overview of the SMIL Timesheets is given in the Overview section. After that, the normative definition of SMIL Timesheets is provided.

The normative definition starts with an introduction of the basic concepts behind SMIL Timesheets. After that, the new elements **timesheet** and **item** and use of CSS selectors are described in the SMIL Timesheet Specific Elements section. Then, the use of SMIL 3.0 Timing and Synchronization modules is discussed in the Timing and Synchronization section. After that, the Prefetch section describes how the SMIL 3.0 PrefetchControl is used. Next, Animation section covers the use of SMIL 3.0 BasicAnimation module. Finally, integration with CSS layout is discussed in the Integration with CSS Layout section.

2 Overview

This section is informative

The following example is a simple slide show. The SMIL Timesheet is located in the head section. It is declared with the **timesheet** element, which belongs to the smil

namespace. In this example, the **seq** element is used for timing. The **item** elements contain references to the body of the document and additional timing information. The elements of the document body are selected using CSS selectors. In this example, the id selectors (i.e., `"#Slide1"`, `"#Slide2"`, and `"#Slide3"`) are used. Based on the timing information a new slide is shown after five seconds.

```
<?xml version="1.0"?>
<html xmlns="http://www.w3.org/1999/xhtml"
         xmlns:smil="http://www.w3.org/ns/SMIL30">

  <head>
    <title>Timesheet Example</title>
    <smil:timesheet>
      <smil:seq>
        <smil:item select="#Slide1" dur="5s" />
        <smil:item select="#Slide2" dur="5s" />
        <smil:item select="#Slide3" dur="5s" />
      </siml:seq>
    </smil:timesheet>
  </head>

  <body>
    <div xml:id ="Slide1" class="Slide">
      <h1 xml:id="Title1">Timesheets</h1>
      <ul>
        <li xml:id="Bullet1_1" class="Bullet">Timesheets defines temporal dimension of documents</li
        <li xml:id="Bullet1_2" class="Bullet">Timesheets uses SMIL Timing and Synchronization primit
        <li xml:id="Bullet1_3" class="Bullet">Timesheet is located in the head section of the docume
        <li xml:id="Bullet1_4" class="Bullet">The body elements are selected using CSS selectors</li
      </ul>
    </div>
    <div xml:id ="Slide2" class="Slide">
      <h1 xml:id="Title2">CSS Selectors</h1>
      <ul>
        <li xml:id="Bullet2_1" class="Bullet">Timesheets uses CSS selectors to reference body elemer
        <li xml:id="Bullet2_2" class="Bullet">Id selector references individual elementst</li>
        <li xml:id="Bullet2_3" class="Bullet">Other selectors can be used to match multiple element
        <li xml:id="Bullet2_4" class="Bullet">For example, class selector matches all elements that
      </ul>
    </div>
    <div xml:id ="Slide3" class="Slide">
      <h1 xml:id="Title3">Animations</h1>
      <ul>
        <li xml:id="Bullet3_1" class="Bullet">Timesheets includes also BasicAnimation module</li>
        <li xml:id="Bullet3_2" class="Bullet">Supported elements are animate, set, animateMotion, ar
        <li xml:id="Bullet3_3" class="Bullet">The animation can be defined with either values or fro
        <li xml:id="Bullet3_4" class="Bullet">The target elements are referenced with CSS selectors
      </ul>
    </div>
  </body>
</html>
```

3 SMIL Timesheets Basic Concepts

This section is normative.

This section defines the four basic concepts behind Timesheets:

- Timing and Synchronization
- Selection Mechanism
- Event Model
- Index Functionality

3.1 Timing and Synchronization

SMIL Timesheets use five SMIL 3.0 Timing and Synchronization modules:

1. BasicInlineTiming
2. EventTiming
3. BasicTimeContainers
4. BasicExclTimeContainers
5. RepeatTiming

The timing semantics of SMIL Timesheets is based on the semantics of the SMIL 3.0 Timing and Synchronization module. Host document's elements are set to `"active"` and `"inactive"` following the semantics of the defined time containers. In case of an element in the document that is not referenced by the Timsheets, the element will be visible as dictated by the stylesheet. By default the element will remain visible all the time.

The base time, or the *syncbase* of a **timesheet** element is the moment when the element is started by its parent. Starting an element does not necessarily make the referenced media element visible. Rather, it sets the time moment `"0s"`, to which the element's attributes are compared.

The syncbase of the child elements of a time container is dependent on the type of the container. The children of **par** and **excl** elements have the starting time of their parent as their syncbase. The children of the **seq** element consider the end time of preceding child as their syncbase. This time is resolved only at the moment the preceding child ends.

This section is informative.

Figure 1 shows a simple example of the semantics of a timesheet. A parallel time container has two children, which are **item** elements referencing two media elements. When the **par** is started, it activates both its children with the current time as their syncbase. The media elements referenced by the children are not activated yet. At time moment *1s*, the media element *item1* is activated, according to the begin value of the referencing **timesheet** element. At *2s*, the *item2* is activated. At *3s*, the duration of *item1* runs out, so it is stopped and the corresponding timesheet element deactivates itself. At *4s*, the parent container stops according to its duration attribute, stopping all of its active children.

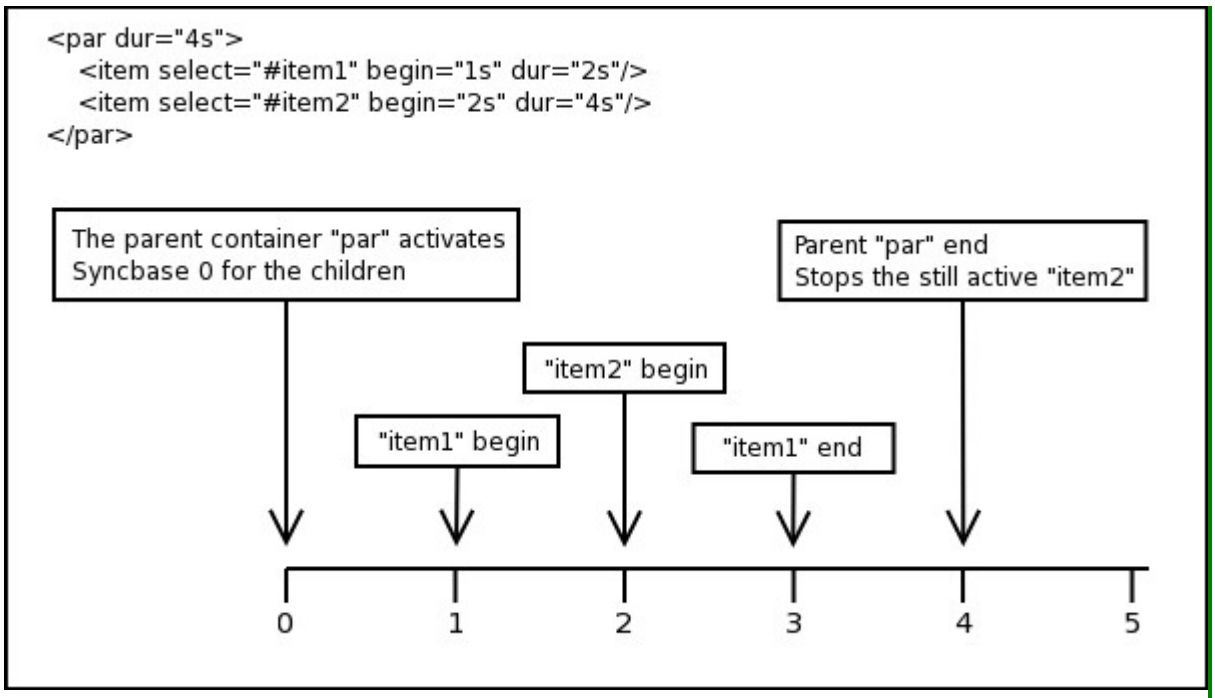

Figure 1: Timeline of a Timesheets

The duration of an element is primarily defined by the **dur** attribute. If the element is not stopped prematurely, due to an event or scheduling of its parent, the **dur** attribute tells the overall duration of the element. The element will not stop until this duration has passed, and will not stay active longer than this duration. The value of the **dur** attribute will prevail:

- when the sum of the durations of the children of a time container extends the value of the attribute;
- when the implicit duration of the media elements, like audio or video is longer than the specified duration;
- when the **repeatCount** or **repeatDur** attributes extend the duration of a media element to be longer than the specified duration.

If the duration is not set, the duration of an element depends on the type of the referenced elements, or the durations of the children. The timesheet items can reference to discrete and continuous elements in the document. The intrisic duration of an **item** element referencing a discrete media element is `"0s"`. But, the default **fill** behaviour is `"freeze"`. Continuous media elements have their own durations, which will be also used as the duration of the timesheet element. The default duration of the time container depends on the duration of their children. By default, a time container ends when the last scheduled child ends. By default the **begin** attibute of a children of a **seq** and a **par** time containers is `"0s"`. On the other hand, the **begin** attribute of a children of a **excl** time container is `"indefinite"`.

3.2 Selection Mechanism

The **item** target element references elements in the document host language(s). It is the mechanism SMIL Timesheets uses to select the elements in the host language

that will be timed.

The **item** can reference a media element in the host language or a set of elements by the element's tag, id or class. If follows the syntax and processing mechanism of the CSS Selectors.

The CSS selector can match more than one element in the document host language, for example, when CSS class selector is used. When multiple matching occur, a sequence of elements that all match exactly the **item** element is constructed. This sequence is ordered following the host document's order and the selection mechanism will consider the elements as ordered in such list. Thus, one **item** element can reference multiple host language elements at the same time.

This section is informative.

In the following example, an **item** element contains **seq** time container and further another **item** element.

```
<smil:timesheet>
  <smil:seq>
    <smil:item xml:id="first" select=".Slide" dur="7s">
      <smil:seq>
        <smil:item xml:id="second" select=".Bullet" dur="1s" />
      </smil:seq>
    </smil:item>
  </smil:seq>
</smil:timesheet>
```

The "first" **item** element selects one slide at a time and shows each slide for 7 seconds. Within each slide the "second" **item** element selects each bullet and shows it for 1 second.

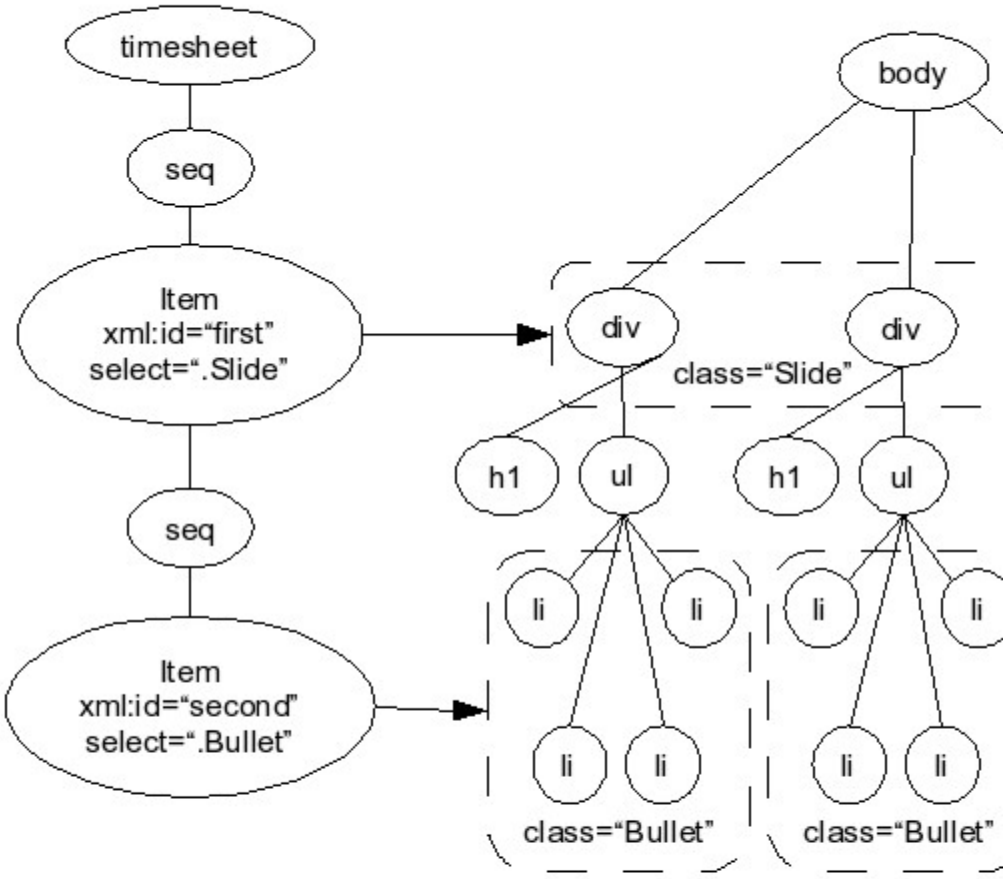

Figure 1: Item Selection Process for Nested Items

The item selection process of the above example is depicted in the Figure 1. The left side of the figure shows the timesheet as a tree, while the right side presents the body of the host document also as a tree. As the figure shows, the "first" item element selects the slides by using the ".Slide" CSS class selector. The selector matches all the div elements of the host document body, since they have the class attribute set to "Slide". Thus, three slides are shown in sequential order.

The "second" item elements uses the ".Bullet" CSS class selector to select all the bullets of the presentation. However, the scope of the selection is limited by the first item element. WIthin the first slide, the CSS selector can match only the

descendants of the div element "Slide1". Similarly within the second slide, the CSS selector can match only the descendants of the div element "Slide2". The same applies also to the third slide.

3.3 Event Model

The **begin**, **dur**, and **end** attributes can contain references to DOM events [DOM2Events]. DOM events can be triggered by user interaction or by some other event in the document. Events are divided into two distinct groups: internal events and user events.

Internal events are dispatched from within the timesheets. They can be used by other elements in the timesheet to create relations between different parts of the timeline. The events specified are `beginEvent` event, which is dispatched when an element starts and `endEvent` event, which is dispatched when an element stops. User events are triggered by user actions.

A **timesheet** element is set to listen to a certain event by specifying the event's target and type by either the **begin**, **dur**, or **end** attributes. When specified by the **begin** attribute, an inactive or started but not yet activated element will be activated when it receives the specified event. The parent time containers and item elements have to be active, though.

When the element is specified to stop according to an event, the element informs its parent that it has stopped and parent then decides what should happen next. Of course, some other element could be waiting to be activated according to the `endEvent` event from the particular element.

3.4 Index Functionality

As defined in the Selection Mechanism section, when multiple elements in the host document match multiple items, an ordered list of item elements is constructed following the document order. In case only one element matches the list will include one element and if there are no matches the list will be empty. SMIL Timesheets provide a way to identify each of the elements by using the `index()` function. Thus, the `index()` function becomes a shortcut to each of the elements composing the ordered list and provide index numbers for the `DOMActivate` event references in the Timesheets.

This section is informative.

The following example gives an example on how the `index()` function works. The example is an image show, which consist of images and corresponding thumbnail images. The thumbnail images can be used to select an individual image for viewing. Only one image is shown at a time.

```
<html xmlns="http://www.w3.org/1999/xhtml"
        xmlns:smil="http://www.w3.org/ns/SMIL30">
  <head>
    <title>Timesheet Example</title>
```

```
      <smil:timesheet>
        <smil:par>

          <smil:excl>
            <smil:item select="#Image1" begin="Thumbnail1.DOMActivate" />
            <smil:item select="#Image2" begin="Thumbnail2.DOMActivate" />
            <smil:item select="#Image3" begin="Thumbnail3.DOMActivate" />
            <smil:item select="#Image4" begin="Thumbnail4.DOMActivate" />
            <smil:item select="#Image5" begin="Thumbnail5.DOMActivate" />

          </smil:excl>
          <smil:par>
            <smil:item select="#Thumbnail1" />
            <smil:item select="#Thumbnail2" />
            <smil:item select="#Thumbnail3" />
            <smil:item select="#Thumbnail4" />

            <smil:item select="#Thumbnail5" />
          </smil:par>
        </smil:par>
      </smil:timesheet>
    </head>
    <body>

      <img alt="image1" src="Image1.jpg" class="Image" xml:id="Image1" />
      <img alt="image2" src="Image2.jpg" class="Image" xml:id="Image2" />
      <img alt="image3" src="Image3.jpg" class="Image" xml:id="Image3" />
      <img alt="image4" src="Image4.jpg" class="Image" xml:id="Image4" />
      <img alt="image5" src="Image5.jpg" class="Image" xml:id="Image5" />
      <br />

      <button class="Thumbnail" xml:id="Thumbnail1">
        <img src="Thumbnail1.jpg" />
      </button>
      <button class="Thumbnail" xml:id="Thumbnail2">
        <img src="Thumbnail2.jpg" />
      </button>

      <button class="Thumbnail" xml:id="Thumbnail3">
        <img src="Thumbnail3.jpg" />
      </button>
      <button class="Thumbnail" xml:id="Thumbnail4">
        <img src="Thumbnail4.jpg" />
      </button>

      <button class="Thumbnail" xml:id="Thumbnail5">
        <img src="Thumbnail5.jpg" />
      </button>
    </body>
  </html>
```

In the above example, the body of the HTML file consists of five pictures and five buttons, which contain thumbnails of the same images. The idea is to show one image at a time. The image to be shown is selected by the thumbnail buttons. Therefore, the timesheets consist of one **par** time container, which contains one **excl** and another **par** time container. Within the **excl** time container the **item** elements select the images, while the **item** elements within the second **par** time container select the individual thumbnail buttons. `DOMActivate` events cause the **excl** time container to change the shown image according to which button was pushed.

The problem with the above example is that each time the host document is changed, for example, by changing an image, deleting an image, or adding a new image, the Timesheet has to be updated. Therefore, the Timesheet is not reusable. The main problem lies in the references to the `DOMActive` events, because the index numbers have to be updated everytime a modification in the host document occurs. Thus, there is need to automatically generate the index numbers for the `DOMActivate` event references in the Timesheets. The `index()` function can be used exactly for this purpose. In the Timesheet below, the `index()` function is used within

the **begin** attribute of **item** element, which selects the images within the **excl** time container.

```
<smil:timesheet>
  <smil:par>
    <smil:excl>
      <smil:item select=".Image" begin="index(Thumbnail).DOMActivate" />
    </smil:excl>
    <smil:par>

      <smil:item select=".Thumbnail" />
    </smil:par>
  </smil:par>
</smil:timesheet>
```

The `index()` function adds the index number to the Thumbnail parameter (e.g., `Thumbnail0.DOMActivate`, `Thumbnail1.DOMActivate`, etc.) in the **begin** attribute. Therefore, one `".Image"` class selector can be used in the **select** attribute instead of severall id selectors. The main advantage of this is that changing the order of images, deleting images, or adding new images does not any longer require updating the Timesheet. Thus, the Timesheets is now much more reusable.

However, the Timesheets can be written in even more compact format. As defined in the Timing and Synchronization section, when an element in the host language does not match any **item** in the Timesheet, the element will remain visible all the time. Since all the thumbnail images in buttons are allways shown, the second **par** is actually unnecessary. When it is removed, also the first **par** time container becomes unnecessary. Therefore, the most simple Timesheets for this use case is as follows.

```
<smil:timesheet>
  <smil:excl>
    <smil:item select=".Image" begin="index(Thumbnail).DOMActivate" />
  </smil:excl>
</smil:timesheet>
```

4 SMIL Timesheet Specific Elements

This section is normative.

This section defines the new **timesheet** and **item** elements and their attributes.

4.1 The **timesheet** element

The **timesheet** element is located in the head section of the document. It defines a parent container for other SMIL Timesheet elements.

Element Attributes

The **timesheet** element defines two attributes: **src** and **media**.

*The **src** Attribute*

The **src** attribute tells the location of an external timesheet. With this attribute a common timesheet can be reused in multiple documents. The attribute must contain a valid URI.

This section is informative

The following example shows how the timing information can be provided as an external document.

```
--- example.xhtml ----
<?xml version="1.0"?>
<html xmlns="http://www.w3.org/1999/xhtml"
         xmlns:smil="http://www.w3.org/ns/SMIL30">

  <head>
    <title>Timesheet Example (External Document)</title>
    <smil:timesheet src="./timesheet.smil" />
  </head>
---   ---

--- timesheet.smil ---
<?xml version="1.0"?>

<smil:timesheet xmlns:smil="http://www.w3.org/ns/SMIL30">
 <smil:seq>
   <smil:item select="#Slide1" dur="5s" />
   <smil:item select="#Slide2" dur="5s" />
   <smil:item select="#Slide3" dur="5s" />
 </smil:seq>
</smil:timesheet>
---   ---
```

This section is informative

In non-XML markup languages, the link element can be used to reference an external timesheet document. In the following example, a HTML document contains a reference to external timesheet.smil document.

```
<!doctype html public "-//W3C//DTD HTML 4.01//EN">
<html>
  <head>
    <link href="timesheet.smil" rel="timesheet" type="application/smil+xml">
  </head>
</html>
```

The language designers of the non-XML host language language should define what the different attributes of the link element mean. For example, the HTML language should define the timesheet link type.

*The **media** Attribute*

The **media** attribute is used for selecting the most suitable timesheet for the current media device. It works in the same way as the @media rule in the CSS stylesheets [CSS2]. The **media** attribute contains a comma separated list of CSS media types. The timesheet timing and animation information is applied to the host language document, only when the target device media type matches one of the media types defined by the **media** attribute. If the media attribute is not defined, the default value is

"all".

This section is informative

In the following example, the media attribute contains only one CSS media type: projection. Therefore, the timing is applied to the slideshow only when the slideshow is given as a presentation, e.g., the browser is in so called full screen mode. Otherwise, the timing information is not applied, and thus all the slides are shown at the same time.

```
<smil:timesheet media="projection">
  <smil:seq>
    <smil:item select="#Slide1" dur="5s" />
    <smil:item select="#Slide2" dur="5s" />
    <smil:item select="#Slide3" dur="5s" />
  </smil:seq>
</smil:timesheet>
```

Element Content

The **timesheet** element contains the **par**, **seq**, and **excl** time containers. The semantics and restrictions are specified in the the time container attributes. In addition, it contains the **item** and **prefetch** elements. Finally, it also contains the animation elements: **animate**, **set**, **animateMotion**, and **animateColor**.

In case of having timing containers inside of **item** elements, their semantics are the same

4.2 The **item** element

The **item** target element references elements in the document host language(s). The **item** element implements the actual connection between the timesheet and the document. It can reference a media element or a set of elements by the element's tag, id or class; the syntax and processing is the same to the CSS selector [CSS2] syntax.

As indicated in the Selection Mechanism, when multiple elements of the host document match an **item** element an ordered list based on the host document order is constructed.

This section is informative.

The following example illustrates how the CSS class selector can be used to select more than one slides at the same time.

```
<smil:timesheet>
  <smil:seq>
    <smil:item select=".Slide" dur="5s"/>
  </smil:seq>
</smil:timesheet>
```

The class selector `".Slide"` selects all the slides at once. Thus, the above timesheet is equivalent to the first example.

Element Attributes

The **item** element defines two attributes: **select** and **beginInc**. In addition, it uses the **begin**, **dur**, **end**, **repeatCount**, and **repeatDur** attributes defined in the SMIL 3.0 Timing and Synchronization module.

*The **select** Attribute*

The **select** attribute links the timesheet to the document. Its value is a comma-separated list of CSS selectors [CSS2]. The attribute follows the same syntax as the CSS selectors, so that the elements can be referenced by their name, id, or class, or a more complex combination of the selectors. If the attribute targets multiple elements in the host document, **item** controls all of them based on the host document order.

*The **beginInc** Attribute*

When the selector matches more than one element, an ordered list of elements is constructed. The **beginInc** attribute increments the **begin** time of each of the elements, but the first one, by the defined value. Note that the **begin** time of the first element is never modified. The **beginInc** attribute has to be a positive integer.

This section is informative.

The following example illustrates how the beginInc can be used to increment the starting times of individual elements.

```
<smil:par>
  <smil:item select=".Bullet" beginInc="1s" />
</smil:par>
```

In practise, the above example could also be expressed as follows.

```
<smil:par>
  <smil:item select="#Bullet1_1" begin="0s" />
  <smil:item select="#Bullet1_2" begin="1s" />
  <smil:item select="#Bullet1_3" begin="2s" />
  <smil:item select="#Bullet1_4" begin="3s" />
```

```
</smil:par>
```

Element Functions

The **item** element defines one function: `index()`.

The Index() Function

As introduced in the Index Functionality section, the `index()` function can be used for identifying a specific element within the sorted list resulting from multiple items matching. Thus, it can be used to automatically generate index numbers for both internal and external events within the **begin**, **dur**, or **end** attributes of the **item**, **animate**, **set**, **animateMotion**, and **animateColor** elements.

The `index()` function has the format:

```
index(<selector>, <indexStart>)
```

where:

- The `<selector>` parameter contains the CSS selector.
- The `<indexStart>` parameter indicates the index start.

The `<selector>` parameter has to be a valid CSS selector, while the `<indexStart>` parameter has to be a positive integer value. By default `<indexStart>` is `"0"`. If there are no elements in the host document matching the CSS selector, the index() function is not called. If there are elements in the host document matching the CSS selector, an ordered list of the elements is constructed. In this case, when an element from the list is activated, the index() function returns the index of such element within the list plus the `<indexStart>` value.

Element Content

The **item** element can contain:

- time containers: **seq**, **par**, and **excl**
- **prefetch** elements
- animation elements: **animate**, **set**, **animateMotion**, and **animateColor**
- and other **item** elements

However, the direct child of the **item** element can only have one child or none. All other children than the first one are ignored. If the **item** element has several descendants they have to be included within a child time container. The timing and synchronization of the child element is controlled by the **item**, as described in the Timing andSynchronization section. Furthermore, the parent **item** element limits the scope of the CSS selectors of the descendant **item**, **prefetch**, **animate**, **set**, **animateMotion**, and **animateColor** elements to match only descendant elements of the host language elements selected by the parent **item** element, as described in the SelectionMechanism section.

5 SMIL Timing and Synchronization Elements

This section is normative.

~~*This section is informative.*~~

~~The SMIL Timesheets uses five SMIL 3.0 Timing and Synchronization modules:~~

1. ~~BasicInlineTiming~~
2. ~~EventTiming~~
3. ~~BasicTimeContainers~~
4. ~~BasicExclTimeContainers~~
5. ~~RepeatTiming~~

~~This section gives futher details on the attributes and elements of the above mentioned modules. The reader is presumed to have read and be familiar with them.~~

~~The timing semantics of the timesheets is based on the semantics of the SMIL 3.0 Timing and Synchronization chapter, although it has been simplified a bit. This may cause the timing model not to be as powerful in expressing timing relations as SMIL's, but it should be more approachable to the authors.~~

~~The base time, or the *syncbase* of a **timesheet** element is the moment when the element is started by its parent. Starting an element does not necessarily make the referenced media element visible. Rather, it sets the time moment "0s", to which the element's attributes are compared.~~

~~The syncbase of the child elements of a time container is dependent on the type of the container. The children of **par** and **excl** elements have the starting time of their parent as their syncbase. The children of the **seq** element consider the end time of preceding child as their syncbase. This time is resolved only at the moment the preceding child ends.~~

~~Figure 2 shows a simple example of the semantics of a timesheet. A parallel time container has two children, which are **item** elements referencing two media elements. When the **par** is started, it activates both its children with the current time as their syncbase. The media elements referenced by the children are not activated yet. At time moment *1s*, the media element *item1* is activated, according to the begin value of the referencing **timesheet** element. At *2s*, the *item2* is activated. At *3s*, the duration of *item1* runs out, so it is stopped and the corresponding timesheet element deactivates itself. At *4s*, the parent container stops according to its duration attribute, stopping all of its active children.~~

~~The duration of an element is primarily defined by the **dur** attribute. If the element is not stopped prematurely, due to an event or scheduling of its parent, the **dur** attribute tells the overall duration of the element. The element will not stop until this duration has passed, and will not stay active longer than this duration. The value of the **dur** attribute will prevail:~~

- when the sum of the durations of the children of a time container extends the value of the attribute;
- when the implicit duration of the media elements, like audio or video is longer than the specified duration;
- when the **repeatCount** or **repeatDur** attributes extend the duration of a media element to be longer than the specified duration.

If the duration is not set, the duration of an element depends on the type of the referenced elements, or the durations of the children. The timesheet items can reference to discrete and continuous elements in the document. Discrete elements don't have implicit durations, and the implicit duration of an **item** element referencing a discrete element is `"indefinite"`. Continuous elements have their own durations, which will be also used as the duration of the timesheet element. The duration of the timecontainer depends on the durations and activations of its children. The **seq** and **par** elements stay active until all of their children have stopped. The implicit duration of the **excl** is always `"indefinite"`.

5.1 Attributes

The SMIL Timesheets includes the basic timing attributes **begin**, **dur**, and **end** as defined in the BasicInlineTiming and EventTiming modules. In addition **fill** and **endsync** attributes are included as defined in the BasicTimeContainers and BasicExclTimeContainers modules. Also, it includes the **repeatCount** and **repeatDur** attributes as defined in the RepeatTiming module. The rest of the SMIL 3.0 Timing and Synchronization attributes are not supported. Finally, the SMIL Timesheets includes four new Timing and Synchronization attributes: **first**, **prev**, **next**, and **last**.

The begin Attribute

The **begin** attribute supports offset and event values, `"indefinite"`, or a semi-colon separated list of values. All other values are not supported. The allowed values and semantics of the **begin** attribute are defined in the SMIL 3.0 Timing and Synchronization module.

The dur Attribute

The **dur** attribute supports the clock values, `"media"`, and `"indefinite"`. The allowed values and semantics of the **dur** attribute are defined in the SMIL 3.0 Timing and Synchronization module.

The end Attribute

The **end** attribute supports offset and event values, `"indefinite"`, or a semi-colon separated list of values. All other values are not supported. The allowed values and semantics of the **end** attribute are defined in the SMIL 3.0 Timing and Synchronization module.

The fill Attribute

The **fill** attribute allows an author to specify that an element should be extended beyond the active duration by freezing the final state of the element. It is defined in the SMIL 3.0 Timing and Synchronization module. Since SMIL Timesheets does not include transitions, the `fill="transition"` value of **fill** attribute is not supported. Also, since the `fillDefault` attribute is not included in the SMIL Timesheets, the `fill="default"` is interpreted the same as `fill="auto"`.

The endsync Attribute

The **endsync** attribute controls the implicit duration of time containers, as a function of their children. It is defined in the SMIL 3.0 Timing and Synchronization module.

The repeatCount Attribute

The **repeatCount** attribute specifies the number of iterations of a simple duration. It is defined in the SMIL 3.0 Timing and Synchronization module.

The repeatDur Attribute

The **repeatDur** attribute specifies the total duration for repeat. It is defined in the SMIL 3.0 Timing and Synchronization module.

The first Attribute

The **first** attribute sets the current active child of a time container `"inactive"`. Then, it selects the first child element and sets it `"active"`. The **first** attribute can only be used for the **excl** time container. The allowed value of the **first** attribute is a DOM event [DOM2Events].

The prev Attribute

The **prev** attribute first checks, whether the current active child is the first child. If not, it sets the current active child of a time container `"inactive"`. Then, it selects the previous child of the time container and sets it `"active"`. The **prev** attribute can only be used for the **excl** time container. The allowed value of the **prev** attribute is a DOM event [DOM2Events]

The next Attribute

The **next** attribute first checks, whether the current active child is the last child. If not, it sets the current active child of a time container `"inactive"`. Then, it selects the next child of the time container and sets it `"active"`. The **next** attribute can only be used for the **excl** time container. The allowed value of the **next** attribute is a DOM event [DOM2Events]

The last Attribute

The **last** attribute sets the current active child of a time container `"inactive"`. Then, it selects the last child of the time container and sets it `"active"`. The **last** attribute can

only be used for the **excl** time container. The allowed value of the **last** attribute is a DOM event [DOM2Events]

5.2 Elements

The SMIL Timesheets includes three elements **par**, **seq**, and **excl** as defined in the BasicTimeContainers and BasicExclTimeContainers modules.

The par Element

The **par** element short for "parallel", defines a simple time grouping in which multiple elements can play back at the same. It is defined in the SMIL 3.0 Timing and Synchronization module.

The seq Element

The **seq** element defines a sequence of elements in which elements play one after the other. Children of a **seq** can never play simultaneously and play strictly in document order, except when a hyperlink traversal targets an earlier child. It is defined in the SMIL 3.0 Timing and Synchronization module.

The excl Element

The **excl** element defines a time container with semantics based upon **par**, but with the additional constraint that only one child element may play at any given time. If any element begins playing while another is already playing, the element that was playing is stopped. In SMIL Timesshets, it is a simplified version of the **excl** element defined in the SMIL 3.0 Timing and Synchronization module, since the BasicPriorityClassContainers module is not included.

There are also four additional attributes that are specified for the **excl** element. They are **first**, **prev**, **next**, and **last**. These are used to select a certain child of the **excl** element, which facilitates the creation of presentations where the user can have control of the progression of the presentation, for example, a picture show.

This section is informative.

The following example illustrates how the additional **first**, **prev**, **next**, and **last** attributes can be used in slideshow. It contains a new timesheet and four buttons, which are appended to the body part of first example.

```
<head>
  <smil:timesheet>
    <smil:excl first="first.DOMActivate" prev="prev.DOMActivate" next="next.DOMActivate" last="las
      <smil:item select=".Slide" />
    </smil:excl>
  </smil:timesheet>
</head>

<body>
  ...
  <button xml:id="first">First slide</button>
  <button xml:id="prev">Previous slide</button>
  <button xml:id="next">Next slide</button>
```

```
    <button xml:id="last">Last slide</button>
  </body>
```

The *first* button selects the first slide, the *prev* button selects the previous slide, the *next* button selects the next slide, and finally the *last* button selects the last slide.

6 Prefetch

This section is normative.

The SMIL Timesheets uses the SMIL 3.0 PrefetchControl module. The module defines one element **prefetch**, which has three attributes **mediaSize**, **mediaTime**, and **bandwidth**. The original SMIL 3.0 PrefetchControl module uses the **src** attribute for locating and fetching the associated media. In the SMIL Timesheets, the **src** attribute is replaced with the **select** attribute. This section gives further details on how the attributes and elements of the SMIL 3.0 PrefetchControl module are used in Timesheets.

If multiple elements in the host language match the **select** attribute, the behaviour is the same as with the **item** element. An oredered list of elements is constructed based on the host document order, so that the parent time container determines whether the prefetching of the elements is done in parallel or in sequence.

6.1 Attributes

The SMIL Timesheets includes all three attributes of the SMIL 3.0 PrefetchControl module: **mediaSize**, **mediaTime**, and **bandwidth**.

The mediaSize Attribute

The **meadiaSize** attribute defines how much of the resource to fetch as a function of the file size of the resource. It is defined in the SMIL 3.0 PrefetchControl module.

The mediaTime Attribute

The **meadiaTime** attribute defines how much of the resource to fetch as a function of the duration of the resource. It is defined in the SMIL 3.0 PrefetchControl module.

The bandwidth Attribute

The **bandwidth** attribute defines how much network bandwidth the user agent should use when doing the prefetch. It is defined in the SMIL 3.0 PrefetchControl module.

6.2 Elements

The SMIL Timesheets includes the **prefetch** element of the SMIL 3.0 PrefetchControl module.

The prefetch Element

The **prefetch** element gives authoring tools or savvy authors the ability to schedule retrieval of resources when they think that there is available bandwidth or time to do it. It is defined in the SMIL 3.0 PrefetchControl module. It uses the **mediaSize**, **mediaTime**, and **bandwidth** attributes.

This section is informative.

The following example illustrates how to use the **prefetch** element.

```
<head>
  <smil:timesheet>
    <smil:seq>
      <smil:prefetch select="#Image1" mediaSize="100%" />
      <smil:par>
        <smil:item select="#Image1" dur="5s" />
        <smil:prefetch select="#Image2" />
      </smil:par>
      <smil:par>
        <smil:item select="#Image2" dur="5s" />
        <smil:prefetch select="#Image3" />
      </smil:par>
      <smil:par>
        <smil:item select="#Image3" dur="5s" />
        <smil:prefetch select="#Image4" />
      </smil:par>
      <smil:par>
        <smil:item select="#Image4" dur="5s" />
        <smil:prefetch select="#Image5" />
      </smil:par>
      <smil:item select="#Image5" dur="5s" />
    </smil:seq>
  </smil:timesheet>
</head>
<body>
  <div xml:id="Images">
   <img src="img/Image1.jpg" alt="Image1" class="Image" xml:id="Image1" />
   <img src="img/Image2.jpg" alt="Image2" class="Image" xml:id="Image2" />
   <img src="img/Image3.jpg" alt="Image3" class="Image" xml:id="Image3" />
   <img src="img/Image4.jpg" alt="Image4" class="Image" xml:id="Image4" />
   <img src="img/Image5.jpg" alt="Image5" class="Image" xml:id="Image5" />
  </div>
</body>
```

7 Animation

This section is normative.

The SMIL Timesheets uses the SMIL 3.0 BasicAnimation module. This section gives further details on how the attributes and elements of the SMIL 3.0 BasicAnimation module are used in Timesheets.The reader is presumed to have read the SMIL 3.0 BasicAnimation module and be familiar with it.

7.1 Attributes

This section is informative.

In SMIL 3.0 BasicAnimation module, the animation target element can be referenced either with `targetElement` or XLink `href` attribute. According to the module, a host language designer should select only one of them. In Timesheets, the `href` is more natural choice. However, use of wider selection of CSS selectors is prefered as in the **item** element. Therefore, the **select** attribute is used instead of `targetElement` and `href` attributes. Because of this choise, the XLink `actuate`, `show`, and `type` attributes are also not used.

An animation element can define the target element of the animation either explicitly or implicitly. An explicit definition uses the **select** attribute to specify the target element. The syntax for this is described below. If no explicit target is specified, that is, the animation element does not specify the **select** attribute, the implicit target element is the host language element or elements referenced by the parent or ancestor **item** element.

Timesheets uses the **select** attribute to specify the target element. Therefore it does not include the `targetElement` and the XLink attributes `href`, `actuate`, `show`, and `type` attributes. The target attribute is specified with the **attributeName** attribute and optional **attributeType** attribute, which specifies whether the target attribute is CSS property or XML attribute. Animation is described either as a list of **values**, or in a simplified form that describes the **from**, **to**, and **by** values. The **calcMode** attribute specifies the interpolation mode for the animation. The cumulative and additive behavior of repeating animations is controlled with the **accumulate** and **additive** attributes, respectively. The **origin** attribute specifies the origin of motion for the animation. The **begin**, **dur**, **end**, and **fill** attributes are used to control the timing of animation, while **repeatCount** and **repeatDur** attributes are used to control the repeat of animation. Finally, **beginInc** attribute can be used to increment the begin time, while the `index()` function can be used to automatically generate index numbers for events.

The **select** and **beginInc** attributes and the `index()` function are described in the SMIL Timesheet Specific Elements section, while the **begin**, **dur**, **end**, **fill**, **repeatCount**, and **repeatDur** attributes are described in the Timing and Synchronization section. The other animation attributes are described below.

The attributeName Attribute

The **attributeName** attribute specifies the name of the target attribute. It is defined in the SMIL 3.0 BasicAnimation module.

The attributeType Attribute

The **attributeType** attribute specifies the namespace in which the target attribute and its associated values are defined. It is defined in the SMIL 3.0 BasicAnimation module.

The values Attribute

The **values** attribute contains a semicolon-separated list of one or more values. It is defined in the SMIL 3.0 BasicAnimation module.

The from Attribute

The **from** attribute specifies the starting value of the animation. It is defined in the SMIL 3.0 BasicAnimation module.

The to Attribute

The **to** attribute specifies the ending value of the animation. It is defined in the SMIL 3.0 BasicAnimation module.

The by Attribute

The **by** attribute Specifies a relative offset value for the animation. It is defined in the SMIL 3.0 BasicAnimation module.

The calcMode Attribute

The **calcMode** attribute specifies the interpolation mode for the animation. It is defined in the SMIL 3.0 BasicAnimation module.

The accumulate Attribute

The **accumulate** attribute controls whether or not the animation is cumulative. It is defined in the SMIL 3.0 BasicAnimation module.

The additive Attribute

The **additive** attribute controls whether or not the animation is additive. It is defined in the SMIL 3.0 BasicAnimation module.

The origin Attribute

The **origin** attribute specifies the origin of motion for the animation. It is defined in the SMIL 3.0 BasicAnimation module.

7.2 Elements

The SMIL Timesheets includes all elements of the SMIL 3.0 BasicAnimation module: **animate**, **set**, **animateMotion**, and **animateColor**.

The animate Element

The **animate** element controls the animation of both CSS properties and XML element attributes. It is defined in the SMIL 3.0 BasicAnimation module. It uses the **select**, **beginInc**, **from**, **to**, **by**, **values**, **calcMode**, **accumulate**, **additive**, **begin**, **dur**, **end**, **fill**, **repeatCount**, and **repeatDur** attributes.

This section is informative.

The following example illustrates how the animate element can be used to animate CSS properties of elements of the host language.

```
<smil:timesheet>
  <smil:seq>
    <smil:item select=".Slide" dur="15s">
      <smil:par>
        <smil:item select=".Bullet" beginInc="3s">
          <smil:animate select=".Bullet" attributeType="CSS"
              attributeName="margin-left" values="200;0" dur="1s" />
        </smil:item>
      </smil:par>
    </smil:item>
  </smil:seq>
</smil:timesheet>
```

The slides are shown in sequential order, while the bullets are shown in parallel within each slide. However, each bullet starts 3 seconds after the previous one. Each bullet flies from right to left for 1 second when shown.

The set Element

The **set** element provides a simple means of setting the value of an attribute for a specified duration. It is defined in the SMIL 3.0 BasicAnimation module. It uses the **select**, **beginInc**, **to**, **begin**, **dur**, **end**, **fill**, **repeatCount**, and **repeatDur** attributes.

The animateMotion Element

The **animateMotion** element moves an element along a path. It is defined in the SMIL 3.0 BasicAnimation module. It uses the **select**, **beginInc**, **from**, **to**, **by**, **values**, **calcMode**, **accumulate**, **additive**, **origin**, **begin**, **dur**, **end**, **fill**, **repeatCount**, and **repeatDur** attributes.

The animateColor Element

The **animateColor** element specifies an animation of a color attribute. It is defined in the SMIL 3.0 BasicAnimation module. It uses the **select**, **beginInc**, **from**, **to**, **by**, **values**, **calcMode**, **accumulate**, **additive**, **begin**, **dur**, **end**, **fill**, **repeatCount**, and **repeatDur** attributes.

7 Event Model

~~*This section is informative.*~~

~~The **begin**, **dur**, and **end** attributes can contain references to DOM events [DOM2Events]. DOM events can be triggered by user interaction or by some other event in the document. Events are divided into two distinct groups, internal events and~~

user events.

Internal Events

This section is informative.

Internal events are dispatched from within the timesheets. They can be used by other elements in the timesheet to create relations between different parts of the timeline. The events specified are `beginEvent` event, which is dispatched when an element starts and `endEvent` event, which is dispatched when element stops.

User Events

This section is informative.

User events are triggered by the actions that user makes. A typical example is that the user activates a link in the document, and thus a `DOMActivate` event is dispatched [DOM2Events].

Event Semantics

This section is informative.

A **timesheet** element is set to listen to a certain event by specifying the event's target and type by either the **begin**, **dur**, or **end** attributes. When specified by the **begin** attribute, an inactive or started but not yet activated element will be activated when it receives the specified event. The parent time containers and item elements have to be active, though.

When the element is specified to stop according to an event, it does not cause that much processing in the timesheet. The element informs its parent that it has stopped and parent then decides what should happen next. Of course, some other element could be waiting to be activated according to the `endEvent` event from the particular element.

Index Function

This section is informative.

The `index()` function can be used to automatically generate index numbers for both internal and external events within **begin**, **dur**, or **end** attributes of item element. The following example gives an example on how `index()` function can be used. The example is an image show, which consist of images and corresponding thumbnail images, which can be used to select an individual image for viewing. Only one image is shown at a time.

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:smil="http://www.w3.org/ns/SMIL30">
  <head>
    <title>Timesheet Example</title>
    <smil:timesheet>
      <smil:par>
        <smil:excl>
          <smil:item select="#Image1" begin="Thumbnail1.DOMActivate" />
          <smil:item select="#Image2" begin="Thumbnail2.DOMActivate" />
          <smil:item select="#Image3" begin="Thumbnail3.DOMActivate" />
          <smil:item select="#Image4" begin="Thumbnail4.DOMActivate" />
          <smil:item select="#Image5" begin="Thumbnail5.DOMActivate" />
        </smil:excl>
        <smil:par>
          <smil:item select="#Thumbnail1" />
          <smil:item select="#Thumbnail2" />
          <smil:item select="#Thumbnail3" />
          <smil:item select="#Thumbnail4" />
          <smil:item select="#Thumbnail5" />
        </smil:par>
      </smil:par>
    </smil:timesheet>
  </head>
  <body>
    <img alt="image1" src="Image1.jpg" class="Image" xml:id="Image1" />
    <img alt="image2" src="Image2.jpg" class="Image" xml:id="Image2" />
    <img alt="image3" src="Image3.jpg" class="Image" xml:id="Image3" />
    <img alt="image4" src="Image4.jpg" class="Image" xml:id="Image4" />
    <img alt="image5" src="Image5.jpg" class="Image" xml:id="Image5" />
    <br />
    <button class="Thumbnail" xml:id="Thumbnail1">
      <img src="Thumbnail1.jpg" />
    </button>
    <button class="Thumbnail" xml:id="Thumbnail2">
      <img src="Thumbnail2.jpg" />
    </button>
    <button class="Thumbnail" xml:id="Thumbnail3">
      <img src="Thumbnail3.jpg" />
    </button>
    <button class="Thumbnail" xml:id="Thumbnail4">
      <img src="Thumbnail4.jpg" />
    </button>
    <button class="Thumbnail" xml:id="Thumbnail5">
      <img src="Thumbnail5.jpg" />
    </button>
  </body>
</html>
```

In the above example, the body of the HTML file consists of five pictures and five buttons, which contain thumbnails of the same images. The idea is to show one image at a time. The image to be shown is selected by the thumbnail buttons. Therefore, the timesheets consist of one **par** time container, which contains one **excl** and another **par** time container. Within the **excl** time container the **item** elements select the images, while the **item** elements within the second **par** time container select the individual thumbnail buttons. `DOMActivate` events cause the **excl** time container to change the shown image according to which button was pushed.

The problem with the above example is that each time the position of an image is changed, deleted, or added to the image show, the Timesheet has to be updated. Therefore, the Timesheet is not reusable. The main problem lies in the references to the `DOMActive` events, because the index numbers have to be updated. Thus, there is

need to automatically generate index numbers for the `DOMActivate` event references in the Timesheets. The `index()` function can be used exactly for this purpose. In the Timesheet below, `index()` function is used within the **begin** attribute of **item** element, which selects the images within the **excl** time container.

```
<smil:timesheet>
  <smil:par>
    <smil:excl>
      <smil:item select=".Image" begin="index(Thumbnail).DOMActivate" />
    </smil:excl>
    <smil:par>
      <smil:item select=".Thumbnail" />
    </smil:par>
  </smil:par>
</smil:timesheet>
```

The `index()` function adds the index number to the Thumbnail parameter (e.g., `Thumbnail0.DOMActivate`, `Thumbnail1.DOMActivate`, etc.) in the **begin** attribute. Therefore, one `".Image"` class selector can be used in the **select** attribute instead of severall id selectors. The main advantage of this is that changing the order of images, deleting images, or adding new images does not any longer require updating the Timesheet. Thus, the Timesheets is now much more reusable. However, there is one problem in the above example. The index numbering starts from 0. It should be corrected to start from 1 by using the indexStart attribute as shown in the example below.

```
<smil:timesheet>
  <smil:par>
    <smil:excl>
      <smil:item select=".Image" indexStart="1" begin="index(Thumbnail).DOMActivate" />
    </smil:excl>
    <smil:par>
      <smil:item select=".Thumbnail" />
    </smil:par>
  </smil:par>
</smil:timesheet>
```

Now, the index numbering starts from 1, and thus the Timesheet functions properly. However, the Timesheets can be written in even more compact format. The reason is that all the thumbnail images in buttons are allways shown, and thus the second **par** is actually unnecessary. When it is removed, also the first **par** time container becomes unnecessary. Therefore, the most simple Timesheets for this use case is as follows.

```
<smil:timesheet>
  <smil:excl>
    <smil:item select=".Image" indexStart="1" begin="index(Thumbnail).DOMActivate" />
  </smil:excl>
</smil:timesheet>
```

8 Integration with CSS Layout

This section is informative.

Since SMIL Timesheets only describes the temporal dimension of the document, it must be integrated with a host language's layout system. It can be integrated, for example, with CSS based layout by affecting the CSS properties of the timed elements. For instance, the CSS display property can be used to control, whether an element is displayed or not. The SMIL Timesheets processor sets the CSS display

property to to "none", when the timed element should not be visible based on the timesheet. According to CSS specification, this causes the element to have no effect on the layout of the document, and thus the element is invisible. At the same time the original value should be stored for later use. When the media element should become visible, the original display value can be restored.

The content authors should be aware that according to the CSS specification the descendant elements of an element, which has display value set to none, are also invisible. Therefore, the content authors should check that all the parent elements of an active elements are also set active in the timesheet if they are referenced in the timesheet.

Finally, the content authors should also be aware that only visible elements can increment CSS counters. Therefore, CSS counters might not work as expected when they are used together with timesheet. One solution is to use CSS attributes instead and define their values in the document or increment the attribute values, e.g., using a scripting language.

Appendix A. References

This section is informative.

E.1 Normative References

[CSS2]
"*Cascading Style Sheets, level 2: CSS2 Specification*", B. Bos, et al., Editors. World Wide Web Consortium, 12 May 1998. This W3C Recommendation is available at http://www.w3.org/TR/1998/REC-CSS2-19980512/.
The latest version is available at http://www.w3.org/TR/REC-CSS2/

E.2 Informative References

[DOM2Events]
"*Document Object Model (DOM) Level2 : Events*", Tom Pixley. World Wide Web Consortium, 13 November 2000. This W3C Recommendation is available at http://www.w3.org/TR/2000/REC-DOM-Level-2-Events-20001113/events.html.
The latest version is available at http://www.w3.org/TR/DOM-Level-2-Events/events.html

[SMIL30]
"*Synchronized Multimedia Integration Language (SMIL 3.0)* ", Dick Bulterman et al., Editors. World Wide Web Consortium, 13 July 2007. This W3C Working Draft is available at http://www.w3.org/TR/2007/WD-SMIL3-20070713/.
The latest version is available at http://www.w3.org/TR/SMIL3/

[SVG]
"*Scalable Vector Graphics (SVG) 1.1 Specification*", Jon Ferraiolo et al., Editors. World Wide Web Consortium, 14 January 2003. This W3C Recommendation is available at http://www.w3.org/TR/2003/REC-SVG11-20030114/.
The latest version is available at http://www.w3.org/TR/SVG/

[TimesheetsTutorial]
"*Timesheets Tutorial*", Petri Vuorimaa. This tutorial is avaialble at http://www.tml.tkk.fi/~pv/timesheets/.

[XForms]
"*XForms 1.0 (Third Edition)*", John M. Boyer, Editor. World Wide Web Consortium, 29 October 2007. This W3C Recommendation is available at http://www.w3.org/TR/2007/REC-xforms-20071029/.
The latest version is available at http://www.w3.org/TR/xforms

[XHTML]
"*XHTML 1.0 The Extensible HyperText Markup Language (Second Edition)* " Steven Pemberton et al., Editors. World Wide Web Consortium, 24 January 2000, revised 1 August 2002. This W3C NOTE is available at http://www.w3.org/TR/2002/REC-xhtml1-20020801/.
The latest version is available at http://www.w3.org/TR/xhtml1/

[XHTMLplusSMIL]
"*XHTML+SMIL Profile*" Debbie Newman, Patrick Schmitz, Aaron Patterson. World Wide Web Consortium, 31 January 2002. This W3C NOTE is available at http://www.w3.org/TR/2002/NOTE-XHTMLplusSMIL-20020131/.
The latest version is available at http://www.w3.org/TR/XHTMLplusSMIL/

Appendix B. Acknowledgements

This document has been prepared by the Synchronized Multimedia Working Group (SYMM WG) of the World Wide Web Consortium.
The SYMM WG which specified SMIL 3.0 included the following individuals:

Dick Bulterman, CWI - Pablo Cesar, CWI - Samuel Cruz-Lara, INRIA - Marisa DeMeglio, DAISY Consortium - Xabiel Garc잦Pa義, Universidad de Oviedo - Luiz Fernando Gomes Soares (Invited Experts) - Eric Hyche, RealNetworks - Jack Jansen, CWI - Hiroshi Kawamura, NRCD - Nabil Laya, INRIA - David Melendi, Universidad de Oviedo, Thierry Michel, W3C - Sjoerd Mullender, CWI - Julien Quint, DAISY Consortium - Petri Vuorimaa, Helsinki University of Technology - Daniel Weck, NRCD - Daniel F. Zucker, Invited Expert.