

CARDIFF UNIVERSITY EXAMINATION PAPER

Academic Year: 1999/2000

Examination Period: Autumn

Examination Paper Number: CM0340

Examination Paper Title: Multimedia

Duration: 2 Hours

Do not turn this page over until instructed to do so by the Senior Invigilator.

Structure of Examination Paper:

There are Three pages and Four questions.

Students to be provided with:

Answer Book(s)

Multimedia information sheets

Instructions to Students:

Attempt THREE questions.

Each question is worth 27 marks.

1. (a) What is meant by the terms *Multimedia* and *Hypermedia*?

Distinguish between these two concepts.

[2]

(b) What is meant by the terms *static* media and *dynamic* media?

Give two examples of each type of media.

[4]

~~(c) What types of functionality need to be provided in order to effectively use a wide variety of media in Multimedia applications? Your answer should briefly address how such functionality can be facilitated in general Multimedia applications.~~

~~[9]~~

~~(d) Different types of media will require different types of supporting operations to provide adequate levels of functionality. For the examples of static and dynamic media given in your answer to part 1(b) briefly discuss what operations are needed to support a wide range of multimedia applications.~~

~~[12]~~

2. (a) Why is file or data compression highly desirable for Multimedia activities?

[2]

(b) Briefly explain, clearly identifying the differences between them, how entropy coding and transform coding techniques work for data compression.

Illustrate your answer with a simple example of each type.

[8]

(c) (i) Show how you would use *Huffman coding* to encode the following set of tokens:

BABACACADADABBCBABEBEDDABEEEEBB

How is this message transmitted when encoded?

[7]

(ii) How many bits are needed to transfer this coded message and what is its Entropy?

[4]

(iii) What amendments are required to this coding technique if data is generated live or is otherwise not wholly available? Show how you could use this modified scheme by adding the tokens ADADA to the above message. [6]

3 (a) What are the major factors to be taken into account when considering storage requirements for Multimedia Systems? [4]

(b) What is RAID technology and what advantages does it offer as a medium for the storage and delivery of large data? [4]

(c) Briefly explain the *eight* levels of RAID functionality. [8]

(d) A digital video file is 40 Mb in size. The disk subsystem has four drives and the controller is designed to support read and write onto each drive, concurrently. The digital video is stored using the *disk striping* concept. A block size of 8 Kb is used for each I/O operation.

(i) What is the performance improvement in *sequentially* reading the complete file when compared to a single drive subsystem in terms of the number of operations performed?

(ii) What is the percentage performance improvement for this system compared to a single drive system? [11]

4 (a) Give a definition of a Multimedia Authoring System. What key features should such a system provide? [2]

(b) What Multimedia Authoring paradigms exist? Describe each paradigm briefly. [8]

(c) You have been asked to provide a Multimedia presentation that can support media in both English and French. You may assume that you have been given a sequence of 10 images and a single 50 second digitised audio soundtrack in both languages. Each image should be mapped over consecutive 5 second fragments of the audio. All images are of the same 500x500 pixel dimension.

Describe, giving suitable code fragments, how you would assemble such a presentation using SMIL. Your solution should cover all aspects of the SMIL presentation. [17]

Multimedia CM0340 Supplementary Exam Material

Synchronized Multimedia Integration Language (SMIL) 1.0 Specification

Abstract

This document specifies version 1 of the Synchronized Multimedia Integration Language (SMIL 1.0, pronounced "smile"). SMIL allows integrating a set of independent multimedia objects into a synchronized multimedia presentation. Using SMIL, an author can

- 1. describe the temporal behavior of the presentation
- 2. describe the layout of the presentation on a screen
- 3. associate hyperlinks with media objects

This specification is structured as follows:

- Section 1 presents the specification approach.
- Section 2 defines the "smil" element.
- Section 3 defines the elements that can be contained in the head part of a SMIL document.
- Section 4 defines the elements that can be contained in the body part of a SMIL document. In particular, this Section defines the time model used in SMIL.

Table of Contents

- [1 Specification Approach](#)
- [2 The smil Element](#)
- [3 The Document Head](#)
 - [3.1 The head Element](#)
 - [3.2 The layout Element](#)
 - [3.3 SMIL Basic Layout Language](#)
 - [3.3.1 The region Element](#)
 - [3.3.2 The root-layout Element](#)
 - [3.4 The meta Element](#)
- [4 The Document Body](#)
 - [4.1 The body Element](#)
 - [4.2 Synchronization Elements](#)
 - [4.2.1 The par Element](#)
 - [4.2.2 The seq Element](#)
 - [4.2.3 Media Object Element](#): The ref, animation, audio, img, video, text and textstream elements
 - [4.2.4 SMIL Time Model](#)
 - [4.2.4.1 Time Model Values](#)
 - [4.2.4.2 Determining Values of Model Values for SMIL 1.0 Elements](#)
 - [4.3 The switch Element](#)
 - [4.4 Test Attributes](#)
 - [4.5 Hyperlinking Elements](#)
 - [4.5.1 The a Element](#)
 - [4.5.2 The anchor Element](#)

1 Specification Approach

SMIL documents are XML 1.0 documents [XML10]. The reader is expected to be familiar with the concepts and terms defined in XML 1.0.

This specification does not rely on particular features defined in URLs that cannot potentially be expressed using URNs. Therefore, the more generic term URI [URI] is used throughout the specification.

The syntax of SMIL documents is defined by the DTD in [Section 5.2](#). The syntax of an attribute value that cannot

be defined using the DTD notation is defined together with the first element using an attribute that can contain the attribute value. The syntax of such attribute values is defined using the Extended Backus-Naur Form (EBNF) defined in the XML 1.0 specification.

An element definition is structured as follows: First, all attributes of the element are defined in alphabetical order. An attribute is defined in the following way: If the attribute is used by an element for the first time in the specification, the semantics of the attribute are defined. If the attribute has already been used by another element, the specification refers to the definition of the attribute in the first element that used it. The definition of element attributes is followed by the definition of any attribute values whose syntax cannot be defined using the DTD notation. The final section in an element definition specifies the element content.

2 The smil Element

Element Attributes

The "smil" element can have the following attribute:

id This attribute uniquely identifies an element within a document. Its value is an XML identifier.

Element Content

The "smil" element can contain the following children:

body Defined in [Section 4.1](#)
head Defined in [Section 3.1](#)

3 The Document Head

3.1 The head Element

The "head" element contains information that is not related to the temporal behavior of the presentation.

Element Attributes

The "head" element can have the following attribute:

id Defined in [Section 2](#)

Element Content

The "head" element can contain the following children:

layout Defined in [Section 3.2](#)
meta Defined in [Section 3.4](#)
switch Defined in [Section 4.3](#)

The "head" element may contain any number of "meta" elements and either a "layout" element or a "switch" element.

3.2 The layout Element

The "layout" element determines how the elements in the document's body are positioned on an abstract rendering surface (either visual or acoustic).

If a document contains no layout element, the positioning of the body elements is implementation-dependent.

A SMIL document can contain multiple alternative layouts by enclosing several layout elements within a "switch" element (defined in [Section 4.3](#)). This can be used for example to describe the document's layout using different layout languages.

The following example shows how CSS2 can be used as alternative to the SMIL basic layout language (defined in [Section 3.3](#)):

```
<smil>
  <head>
    <switch>
      <layout type="text/css">
        [region="r"] { top: 20px; left: 20px }
      </layout>
      <layout>
        <region id="r" top="20" left="20" />
      </layout>
    </switch>
  </head>
  <body>
    <seq>
      
    </seq>
  </body>
</smil>
```

(note that in this example, both layout alternatives result in the same layout)

Element Attributes

id	Defined in Section 2
type	This attribute specifies which layout language is used in the layout element. If the player does not understand this language, it must skip all content up until the next "</layout>" tag. The default value of the type attribute is "text/smil-basic-layout".

Element Content

If the type attribute of the layout element has the value "text/smil-basic-layout", it can contain the following elements:

region	Defined in Section 3.3.1
root-layout	Defined in Section 3.3.2

If the type attribute of the "layout" element has another value, the element contains character data.

3.3 SMIL Basic Layout Language

This section defines a basic layout language for SMIL. SMIL basic layout is consistent with the visual rendering model defined in CSS2, it reuses the formatting properties defined by the CSS2 specification, and newly introduces the "fit" attribute [\[CSS2\]](#). The reader is expected to be familiar with the concepts and terms defined in CSS2.

SMIL basic layout only controls the layout of media object elements (defined in [Section 4.2.3](#)). It is illegal to use SMIL basic layout for other SMIL elements.

The type identifier for SMIL basic layout is "text/smil-basic-layout".

Fixed Property Values

The following stylesheet defines the values of the CSS2 properties "display" and "position" that are valid in SMIL basic layout. These property values are fixed:

```
a          (display:block)
anchor     (display:block)
animation  {display: block;
            position: absolute}
body       (display: block)
head       (display: none)
img        (display: block;
            position: absolute)
layout     (display: none)
meta       (display: none)
par        (display: block)
region     (display: none)
ref        (display: block;
            position: absolute)
root-layout (display: none)
seq        (display: block)
smil       (display: block)
switch     (display:block)
text       (display: block;
            position: absolute)
textstream (display: block;
            position: absolute)
video      (display: block;
            position: absolute)
```

Note that as a result of these definitions, all absolutely positioned elements (animation, img, ref, text, textstream and video) are contained within a single containing block defined by the content content edge of the root element (smil).

Default Values

SMIL basic layout defines default values for all layout-related attributes. These are consistent with the initial values of the corresponding properties in CSS2.

If the author wants to select the default layout values for *all* media object elements in a document, the document must contain an empty layout element of type "text/smil-basic-layout" such as:

```
<layout type="text/smil-basic-layout"></layout>
```

3.3.1 The region Element

The region element controls the position, size and scaling of media object elements.

In the following example fragment, the position of a text element is set to a 5 pixel distance from the top border of the rendering window:

```
<smil>
  <head>
    <layout>
      <region id="a" top="5" />
    </layout>
  </head>
  <body>
    <text region="a" src="text.html" dur="10s" />
  </body>
</smil>
```

Element Attributes

The "region" element can have the following attributes:

background-color
The use and definition of this attribute are identical to the "background-color" property in the CSS2 specification, except that SMIL basic layout does not require support for "system colors".
If the background-color attribute is absent, the background is transparent.

fit

This attribute specifies the behavior if the intrinsic height and width of a visual media object differ from the values specified by the height and width attributes in the "region" element. This attribute does not have a 1-1 mapping onto a CSS2 property, but can be simulated in CSS2.

This attribute can have the following values:

- fill
- Scale the object's height and width independently so that the content just touches all edges of the box.
- hidden
- If the intrinsic height (width) of the media object element is smaller than the height (width) defined in the "region" element, render the object starting from the top (left) edge and fill up the remaining height (width) with the background color.

If the intrinsic height (width) of the media object element is greater than the height (width) defined in the "region" element, render the object starting from the top (left) edge until the height (width) defined in the "region" element is reached, and clip the parts of the object below (right of) the height (width).
- meet
- Scale the visual media object while preserving its aspect ratio until its height or width is equal to the value specified by the height or width attributes, while none of the content is clipped. The object's left top corner is positioned at the top-left coordinates of the box, and empty space at the left or bottom is filled up with the background color.
- scroll
- A scrolling mechanism should be invoked when the element's rendered contents exceed its bounds.
- slice
- Scale the visual media object while preserving its aspect ratio so that its height or width are equal to the value specified by the height and width attributes while some of the content may get clipped. Depending on the exact situation, either a horizontal or a vertical slice of the visual media object is displayed. Overflow width is clipped from the right of the media object. Overflow height is clipped from the bottom of the media object.

The default value of "fill" is "hidden".

- height
- The use and definition of this attribute are identical to the "height" property in the CSS2 specification. Attribute values can be "percentage" values, and a variation of the "length" values defined in CSS2. For "length" values, SMIL basic layout only supports pixel units as defined in CSS2. It allows to leave out the "px" unit qualifier in pixel values (the "px" qualifier is required in CSS2).
- id
- Defined in [Section 2](#)

A region element is applied to a positionable element by setting the [region](#) attribute of the positionable element to the id value of the region.

The "id" attribute is required for "region" elements.
- left
- The use and definition of this attribute are identical to the "left" property in the CSS2 specification. Attribute values have the same restrictions as the attribute values of the "height" attribute. The default value is zero.
- skip-content
- This attribute is introduced for future extensibility of SMIL (see [Appendix](#)). It is interpreted in the following two cases:

If a new element is introduced in a future version of SMIL, and this element allows SMIL 1.0 elements as element content, the "skip-content" attribute controls whether this content is processed by a SMIL 1.0 player.

If an empty element in SMIL version 1.0 becomes non-empty in a future SMIL version, the "skip-content" attribute controls whether this content is ignored by a SMIL 1.0 player, or results in a syntax error.

If the value of the "skip-content" attribute is "true", and one of the cases above apply, the content of the element is ignored. If the value is "false", the content of the element is processed. The default value for "skip-content" is "true".

- title
- This attribute offers advisory information about the element for which it is set. Values of the title attribute may be rendered by user agents in a variety of ways. For instance, visual browsers frequently display the title as a "tool tip" (a short message that appears when the pointing device pauses over an object). It is strongly recommended that all "region" elements have a "title" attribute with a meaningful description. Authoring tools should ensure that no element can be introduced into a SMIL document

without this attribute.

- top
- The use and definition of this attribute are identical to the "top" property in the CSS2 specification. Attribute values have the same restrictions as the attribute values of the "height" attribute. The default value is zero.
- width
- The use and definition of this attribute are identical to the "width" property in the CSS2 specification. Attribute values have the same restrictions as the attribute values of the "height" attribute.
- z-index
- The use and definition of this attribute are identical to the "z-index" property in the CSS2 specification, with the following exception:

If two boxes generated by elements A and B have the same stack level, then

1. If the display of an element A starts later than the display of an element B, the box of A is stacked on top of the box of B (temporal order).

2. If the display of the elements starts at the same time, and an element A occurs later in the SMIL document text than an element B, the box of A is stacked on top of the box of B (document tree order as defined in CSS2).

Element Content

"region" is an empty element.

3.3.2 The root-layout element

The "root-layout" element determines the value of the layout properties of the root element, which in turn determines the size of the viewport, e.g. the window in which the SMIL presentation is rendered.

If a document contains more than one "root-layout" element, this is an error, and the document should not be displayed.

Element Attributes

The "root-layout" element can have the following attributes:

- background-color
- Defined in [Section 3.3.1](#)
- height
- Defined in [Section 3.3.1](#)

Sets the height of the root element. Only length values are allowed.
- id
- Defined in [Section 2](#)
- skip-content
- Defined in [Section 3.3.1](#)
- title
- Defined in [Section 3.3.1](#)
- width
- Defined in [Section 3.3.1](#)

Sets the width of the root element. Only length values are allowed.

Element Content

"root-layout" is an empty element.

3.4 The meta Element

The "meta" element can be used to define properties of a document (e.g., author, expiration date, a list of key words, etc.) and assign values to those properties. Each "meta" element specifies a single property/value pair.

Element Attributes

The "meta" element can have the following attributes:

- content
- file:///Main%202Gb/Textures/LECTURE%20STUFF/Multimedia/SMIL/

	This attribute specifies the value of the property defined in the meta element. The "content" attribute is required for "meta" elements.
id	Defined in Section 2
name	This attribute identifies the property defined in the meta element. The "name" attribute is required for "meta" elements.
skip-content	Defined in Section 3.3.1
The list of properties is open-ended. This specification defines the following properties:	
base	The value of this property determines the base URI for all relative URIs used in the document.
pics-label or PICS-Label	The value of this property specifies a valid rating label for the document as defined by PICS [PICS].
title	The value of this property contains the title of the presentation.

Element Content

"meta" is an empty element.

4 The Document Body

4.1 The body Element

The "body" element contains information that is related to the temporal and linking behavior of the document. It implicitly defines a "seq" element (defined in Section 4.2.2, see Section 4.2.4 for a definition of the temporal semantics of the "body" element).

Element Attributes

The "body" element can have the following attribute:

id	Defined in Section 2
----	--------------------------------------

Element Content

The "body" element can contain the following children:

a	Defined in Section 4.5.1
animation	Defined in Section 4.2.3
audio	Defined in Section 4.2.3
img	Defined in Section 4.2.3
par	Defined in Section 4.2.1
ref	Defined in Section 4.2.3
seq	Defined in Section 4.2.2
switch	Defined in Section 4.3
text	Defined in Section 4.2.3
textstream	Defined in Section 4.2.3
video	

Defined in [Section 4.2.3](#)

4.2 Synchronization Elements

4.2.1 The par Element

The children of a par element can overlap in time. The textual order of appearance of children in a par has no significance for the timing of their presentation.

Element Attributes

The "par" element can have the following attributes:

abstract	A brief description of the content contained in the element.
author	The name of the author of the content contained in the element.
begin	This attribute specifies the time for the explicit begin of an element. See Section 4.2.4 for a definition of its semantics. The attribute can contain the following two types of values: delay-value A delay value is a clock-value measuring presentation time. Presentation time advances at the speed of the presentation. It behaves like the timecode shown on a counter of a tape-deck. It can be stopped, decreased or increased either by user actions, or by the player itself. The semantics of a delay value depend on the element's first ancestor that is a synchronization element (i.e. ancestors that are "a" or "switch" elements are ignored): <ul style="list-style-type: none">□ If this ancestor is a "par" element, the value defines a delay from the effective begin of that element (see Figure 4.1).□ If this ancestor is a "seq" element (defined in Section 4.2.2), the value defines a delay from the effective end of the first lexical predecessor that is a synchronization element (see Figure 4.2).
event-value	The element begins when a certain event occurs (see Figure 4.3). Its value is an element-event (see Definition below). The element generating the event must be "in scope". The set of "in scope" elements S is determined as follows: <ol style="list-style-type: none">1. Take all children from the element's first ancestor that is a synchronization element and add them to S.2. Remove all "a" and "switch" elements from S. Add the children of all "a" elements to S, unless they are "switch" elements.

The resulting set S is the set of "in scope" elements.

copyright	The copyright notice of the content contained in the element.
dur	This attribute specifies the explicit duration of an element. See Section 4.2.4 for a definition of its semantics. The attribute value can be a clock value, or the string "indefinite".
end	This attribute specifies the explicit end of an element. See Section 4.2.4 for a definition of its semantics. The attribute can contain the same types of attribute values as the "begin" attribute.
endsync	For a definition of the semantics of this attribute, see Section 4.2.4 . The attribute can have the following values: <ul style="list-style-type: none">o first For a definition of the semantics of this value, see Section 4.2.4.o id-ref This attribute value has the following syntax: id-ref ::= "id(" id-value ")" where "id-value" must be a legal XML identifier. For a definition of the semantics of this value, see Section 4.2.4.o last

For a definition of the semantics of this value, see [Section 4.2.4](#).

The default value of "endsync" is "last".

id	Defined in Section 2
region	<p>This attribute specifies an abstract rendering surface (either visual or acoustic) defined within the layout section of the document. Its value must be an XML identifier. If no rendering surface with this id is defined in the layout section, the values of the formatting properties of this element are determined by the default layout.</p> <p>The "region" attribute on "par" elements cannot be used by the basic layout language for SMIL defined in this specification. It is added for completeness, since it may be required by other layout languages.</p>
repeat	For a definition of the semantics of this attribute, see Section 4.2.4 . The attribute value can be an integer, or the string "indefinite". The default value is 1.
system-bitrate	Defined in Section 4.4
system-captions	Defined in Section 4.4
system-language	Defined in Section 4.4
system-overdub-or-caption	Defined in Section 4.4
system-required	Defined in Section 4.4
system-screen-size	Defined in Section 4.4
system-screen-depth	Defined in Section 4.4
title	<p>Defined in Section 3.3.1</p> <p>It is strongly recommended that all "par" elements have a "title" attribute with a meaningful description. Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.</p>

Note on Synchronization between Children

The accuracy of synchronization between the children in a parallel group is implementation-dependent. Take the example of synchronization in case of playback delays, i.e. the behavior when the "par" element contains two or more continuous media types such as audio or video, and one of them experiences a delay. A player can show the following synchronization behaviors:

hard synchronization	The player synchronizes the children in the "par" element to a common clock (see Figure 4.4 a)).
soft synchronization	Each child of the "par" element has its own clock, which runs independently of the clocks of other children in the "par" element (see Figure 4.4 b)).

Attribute Values

clock value	Clock values have the following syntax:																						
	<table><tr><td>Clock-val</td><td>::= Full-clock-val Partial-clock-val Timecount-val</td></tr><tr><td>Full-clock-val</td><td>::= Hours ":" Minutes ":" Seconds ("." Fraction)?</td></tr><tr><td>Partial-clock-val</td><td>::= Minutes ":" Seconds ("." Fraction)?</td></tr><tr><td>Timecount-val</td><td>::= Timecount ("." Fraction)?</td></tr><tr><td></td><td>{ "h" "min" "s" "ms" }? ; default is "s"</td></tr><tr><td>Hours</td><td>::= 2DIGIT; any positive number</td></tr><tr><td>Minutes</td><td>::= 2DIGIT; range from 00 to 59</td></tr><tr><td>Seconds</td><td>::= 2DIGIT; range from 00 to 59</td></tr><tr><td>Fraction</td><td>::= DIGIT+</td></tr><tr><td>Timecount</td><td>::= DIGIT+</td></tr><tr><td>2DIGIT</td><td>::= DIGIT DIGIT</td></tr></table>	Clock-val	::= Full-clock-val Partial-clock-val Timecount-val	Full-clock-val	::= Hours ":" Minutes ":" Seconds ("." Fraction)?	Partial-clock-val	::= Minutes ":" Seconds ("." Fraction)?	Timecount-val	::= Timecount ("." Fraction)?		{ "h" "min" "s" "ms" }? ; default is "s"	Hours	::= 2DIGIT; any positive number	Minutes	::= 2DIGIT; range from 00 to 59	Seconds	::= 2DIGIT; range from 00 to 59	Fraction	::= DIGIT+	Timecount	::= DIGIT+	2DIGIT	::= DIGIT DIGIT
Clock-val	::= Full-clock-val Partial-clock-val Timecount-val																						
Full-clock-val	::= Hours ":" Minutes ":" Seconds ("." Fraction)?																						
Partial-clock-val	::= Minutes ":" Seconds ("." Fraction)?																						
Timecount-val	::= Timecount ("." Fraction)?																						
	{ "h" "min" "s" "ms" }? ; default is "s"																						
Hours	::= 2DIGIT; any positive number																						
Minutes	::= 2DIGIT; range from 00 to 59																						
Seconds	::= 2DIGIT; range from 00 to 59																						
Fraction	::= DIGIT+																						
Timecount	::= DIGIT+																						
2DIGIT	::= DIGIT DIGIT																						
	file:///Main%202Gb/Textures/LECTURE%20STUFF/Multimedia/SMIL/																						

DIGIT ::= {0-9}

The following are examples of legal clock values:

- o Full clock value: 02:30:03 = 2 hours, 30 minutes and 3 seconds
- o Partial clock value: 02:33 = 2 minutes and 33 seconds
- o Timecount values:
 - 3h = 3 hours
 - 45min = 45 minutes
 - 30s = 30 seconds
 - 5ms = 5 milliseconds

A fraction x with n digits represents the following value:

x * 1/10**n

Examples:

00.5s = 5 * 1/10 seconds = 500 milliseconds
00:00.005 = 5 * 1/1000 seconds = 5 milliseconds

element-event value
An *element event* value specifies a particular event in a synchronization element.
An element event has the following syntax:

Element-event	::= "id(" Event-source ")(" Event ")"
Event-source	::= Id-value
Event	::= "begin" Clock-val "end"

The following events are defined:

begin	<p>This event is generated at an element's effective begin.</p> <p>Example use: begin="id(x) (begin) "</p>
clock-val	<p>This event is generated when a clock associated with an element reaches a particular value. This clock starts at 0 at the element's effective begin. For "par" and "seq" elements, the clock gives the presentation time elapsed since the effective begin of the element. For media object elements, the semantics are implementation-dependent. The clock may either give presentation time elapsed since the effective begin, or it may give the media time of the object. The latter may differ from the presentation time that elapsed since the object's display was started e.g. due to rendering or network delays, and is the recommended approach.</p> <p>It is an error to use a clock value that exceeds the value of the effective duration of the element generating the event.</p>
	Example use: begin="id(x) (45s) "
end	<p>This event is generated at the element's effective end.</p> <p>Example use: begin="id(x) (end) "</p>

Element Content

The par element can contain the following children:

a	Defined in Section 4.5.1
animation	Defined in Section 4.2.3
audio	Defined in Section 4.2.3
img	Defined in Section 4.2.3
par	

ref	Defined in Section 4.2.1
seq	Defined in Section 4.2.3
switch	Defined in Section 4.2.2
text	Defined in Section 4.3
textstream	Defined in Section 4.2.3
video	Defined in Section 4.2.3

All of these elements may appear multiple times as direct children of a par element.

4.2.2 The seq Element

The children of a "seq" element form a temporal sequence.

Attributes

The seq element can have the following attributes:

abstract	Defined in Section 4.2.1
author	Defined in Section 4.2.1
begin	Defined in Section 4.2.1
copyright	Defined in Section 4.2.1
dur	Defined in Section 4.2.1
end	Defined in Section 4.2.1
id	Defined in Section 2
region	Defined in Section 4.2.1 The region attribute on "seq" elements cannot be used by the basic layout language for SMIL defined in this specification. It is added for completeness, since it may be required by other layout languages.
repeat	Defined in Section 4.2.1
system-bitrate	Defined in Section 4.4
system-captions	Defined in Section 4.4
system-language	Defined in Section 4.4
system-overdub-or-caption	Defined in Section 4.4
system-required	Defined in Section 4.4
system-screen-size	Defined in Section 4.4
system-screen-depth	Defined in Section 4.4
title	Defined in Section 3.3.1 It is strongly recommended that all "seq" elements have a "title" attribute with a meaningful description. Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.

Element Content

The seq element can contain the following children:

a	Defined in Section 4.5.1
animation	Defined in Section 4.2.3
audio	Defined in Section 4.2.3
img	Defined in Section 4.2.3
par	Defined in Section 4.2.1
ref	Defined in Section 4.2.3
seq	Defined in Section 4.2.2
switch	Defined in Section 4.3
text	Defined in Section 4.2.3
textstream	Defined in Section 4.2.3
video	Defined in Section 4.2.3

4.2.3 Media Object Elements: The ref, animation, audio, img, video, text and textstream elements

The media object elements allow the inclusion of media objects into a SMIL presentation. Media objects are included by reference (using a URI).

There are two types of media objects: media objects with an intrinsic duration (e.g. video, audio) (also called "continuous media"), and media objects without intrinsic duration (e.g. text, image) (also called "discrete media").

Anchors and links can be attached to visual media objects, i.e. media objects rendered on a visual abstract rendering surface.

When playing back a media object, the player must not derive the exact type of the media object from the name of the media object element. Instead, it must rely solely on other sources about the type, such as type information contained in the "type" attribute, or the type information communicated by a server or the operating system.

Authors, however, should make sure that the group into which of the media object falls (animation, audio, img, video, text or textstream) is reflected in the element name. This is in order to increase the readability of the SMIL document. When in doubt about the group of a media object, authors should use the generic "ref" element.

Element Attributes

Media object elements can have the following attributes:

abstract	Defined in Section 4.2.1
alt	For user agents that cannot display a particular media-object, this attribute specifies alternate text. It is strongly recommended that all media object elements have an "alt" attribute with a meaningful description. Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.
author	Defined in Section 4.2.1
begin	Defined in Section 4.2.1
clip-begin	

The clip-begin attribute specifies the beginning of a sub-clip of a continuous media object as offset from the start of the media object. Values in the clip-begin attribute have the following syntax:

```
Clip-time-value ::= Metric "=" ( Clock-val | Smpte-val )
Metric          ::= Smpte-type | "npt"
Smpte-type      ::= "smpte" | "smpte-30-drop" | "smpte-25"
Smpte-val       ::= Hours ":" Minutes ":" Seconds
                 { ":" Frames { "." Subframes } }
Hours           ::= 2DIGIT
Minutes         ::= 2DIGIT
Seconds         ::= 2DIGIT
Frames          ::= 2DIGIT
Subframes       ::= 2DIGIT
```

The value of this attribute consists of a metric specifier, followed by a time value whose syntax and semantics depend on the metric specifier. The following formats are allowed:

SMPTE Timestamp
SMPTE time codes [SMPTE] can be used for frame-level access accuracy. The metric specifier can have the following values:

- smpte
- smpte-30-drop
These values indicate the use of the "SMPTE 30 drop" format with 29.97 frames per second. The "frames" field in the time value can assume the values 0 through 29. The difference between 30 and 29.97 frames per second is handled by dropping the first two frame indices (values 00 and 01) of every minute, except every tenth minute.
- smpte-25
The "frames" field in the time specification can assume the values 0 through 24.

The time value has the format hours:minutes:seconds:frames.subframes. If the frame value is zero, it may be omitted. Subframes are measured in one-hundredth of a frame. Examples:

```
clip-begin="smpte=10:12:33:20"
```

Normal Play Time
Normal Play Time expresses time in terms of SMIL clock values. The metric specifier is "npt", and the syntax of the time value is identical to the syntax of SMIL clock values.

```
Examples:
clip-begin="npt=123.45s"
clip-begin="npt=12:05:35.3"
```

clip-end
The clip-end attribute specifies the end of a sub-clip of a continuous media object (such as audio, video or another presentation) that should be played. It uses the same attribute value syntax as the clip-begin attribute. If the value of the "clip-end" attribute exceeds the duration of the media object, the value is ignored, and the clip end is set equal to the effective end of the media object.

copyright
Defined in [Section 4.2.1](#)

dur
Defined in [Section 4.2.1](#)

end
Defined in [Section 4.2.1](#)

fill
For a definition of the semantics of this attribute, see Section 4.2.4. The attribute can have the values "remove" and "freeze".

id
Defined in [Section 2](#)

longdesc
This attribute specifies a link (URI) to a long description of a media object. This description should supplement the short description provided using the alt attribute. When the media-object has associated anchors, this attribute should provide information about the anchor's contents.

region
Defined in [Section 4.2.1](#)

src
The value of the src attribute is the URI of the media object.

system-bitrate

Defined in [Section 4.4](#)
system-captions

Defined in [Section 4.4](#)
system-language

Defined in [Section 4.4](#)
system-overdub-or-caption

Defined in [Section 4.4](#)
system-required

Defined in [Section 4.4](#)
system-screen-size

Defined in [Section 4.4](#)
system-screen-depth

Defined in [Section 4.4](#)
title

Defined in [Section 3.3.1](#)
It is strongly recommended that all media object elements have a "title" attribute with a meaningful description. Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.

type
MIME type of the media object referenced by the "src" attribute.

Element Content

Media Object Elements can contain the following element:

anchor
Defined in [Section 4.5.2](#)

4.2.4 SMIL Time Model

4.2.4.1 Time Model Values

In the following discussion, the term "element" refers to synchronization elements only.

For each element we define the implicit, explicit, desired, and effective begin, duration, and end.

The effective begin/duration/end specify what the reader of the document will perceive.

The implicit, explicit, and desired values are auxiliary values used to define the effective values.

The rules for calculating each of these values for the elements defined in SMIL 1.0 are described in the next section.

1. Each element in SMIL has an *implicit begin*.
2. Each element can be assigned an *explicit begin* by adding a "begin" attribute to the element:

```
begin = "value of explicit-begin"
```

- It is an error if the explicit begin is earlier than the implicit begin of the element.
3. Each element in SMIL has an *implicit end*.
 4. Each element can be assigned an *explicit end* by adding an "end" attribute to the element:

```
end = "value of explicit-end"
```

5. The *implicit duration* of an element is the difference between the implicit end and the implicit begin.
6. Each element in SMIL can be assigned an *explicit duration* by adding a "dur" attribute to the element:

```
dur = "value of explicit-duration"
```

7. The *desired begin* of an element is equal to the explicit begin if one is given, otherwise the desired begin is equal to the implicit begin.
8. Each element has a *desired end*.
9. The *desired duration* of an element is the difference between the desired end and the desired begin.
10. Each element has an *effective begin*.

11. Each element has an *effective end* . (Note: the effective end of a child element can never be later than the effective end of its parent.)
12. The *effective duration* of an element is the difference between the effective end and the effective begin.

4.2.4.2 Determining Time Model Values for SMIL 1.0 Elements

This section defines how time model values are calculated for the synchronization elements of SMIL 1.0 in cases that are not covered by the rules in [Section 4.2.4.1](#).

Determining the *implicit begin* of an element

- The implicit begin of the first child of the "body" element is when the document starts playing. When this is falls outside the scope of this document.
- The implicit begin of a child of a "par" element is equal to the effective begin of the "par" element.
- The implicit begin of the first child of a "seq" element is equal to the effective begin of the "seq" element.
- The implicit begin of any other child of a "seq" element is equal to the desired end time of the previous child of the "seq" element.

Determining the *implicit end* of an element

The first description that matches the element is the one that is to be used:

- An element with a "repeat" attribute with value "indefinite" has an implicit end immediately after its effective begin.
- An element with a "repeat" attribute with a value other than "indefinite" has an implicit end equal to the implicit end of a seq element with the stated number of copies of the element without "repeat" attribute as children.
- A media object element referring to a continuous media object has an implicit end equal to the sum of the effective begin of the element and the intrinsic duration of the media object.
- A media object element referring to a discrete media object such as text or image has an implicit end immediately after its effective begin.
- A "seq" element has an implicit end equal to the desired end of its last child.
- A "par" element has an implicit end that depends on the value of the "endsync" attribute. The implicit end is equal to the sum of the effective begin of the "par" element and the implicit duration which is derived as follows:
 - If the value of the "endsync" attribute is "last", or if the "endsync" attribute is missing, the implicit duration of the "par" element is the maximum of the desired durations of its children.
 - If the value of the "endsync" attribute is "first", the implicit duration of the "par" element is the minimum of the desired durations of its children.
 - If the value of the "endsync" attribute is an id-ref, the implicit duration of the "par" element is equal to the desired duration of the child referenced by the "id-ref".

Determining the *desired end* of an element

- If the element has both an explicit duration and an explicit end, the desired end is the minimum of:
 - the sum of the desired begin and the explicit duration; and
 - the explicit end.
- If the element has an explicit duration but no explicit end, the desired end is the sum of the desired begin and the explicit duration.
- If the element has an explicit end but no explicit duration, the desired end is equal to the explicit end
- Otherwise, the desired end is equal to the implicit end.

Determining the *desired begin* of an element

The desired begin of an element is determined by using rule 7 in [Section 4.2.4.1](#).

Determining the *effective begin* of an element

The *effective begin* of an element is equal to the desired begin of the element, unless the effective end of the parent element is earlier than this time, in which case the element is not shown at all.

Determining the *effective end* of an element

- The effective end of the last child of the body element is player-dependent. The effective end is at least as

- late as the desired end, but whether it is any later is implementation-dependent.
- The effective end of the child of a "par" element can be derived as follows:
 - If the child has a "fill" attribute, and the value of the "fill" attribute is "freeze", the effective end of the child element is equal to the effective end of the parent.
The last state of the element is retained on the screen until the effective end of the element.
 - If the child has a "fill" attribute, and the value of the "fill" attribute is "remove", the effective end of the child element is the minimum of the effective end of the parent and the desired end of the child element.
 - If the child element has no "fill" attribute, the effective end of the child depends on whether or not the child has an explicit duration or end.
 - If the child has an explicit duration or end, the effective end is determined as if the element had a "fill" attribute with value "remove".
 - If the child has neither an explicit duration nor an explicit end, the effective end is determined as if the element had a "fill" attribute with value "freeze".
 - The effective end of the last child of a "seq" element is derived in the same way as the effective end of a child of a "par" element.
 - The effective end of any other child of a "seq" element can be derived as follows:
 - If the child has a "fill" attribute, and the value of the "fill" attribute is "freeze", the effective end of the child element is equal to the effective begin of the next element
 - If the child has a "fill" attribute, and the value of the "fill" attribute is "remove", the effective end of the child element is the minimum of the effective begin of the next element and the desired end of the next child element.
 - If the child element has no "fill" attribute, the effective end of the child depends on whether or not the child has an explicit duration or end.
 - If the child has an explicit duration or end, the effective end is determined as if the element had a fill attribute with value "remove".
 - If the child has neither an explicit duration nor an explicit end, the effective end is determined as if the element had a fill attribute with value "freeze".

4.3 The `switch` Element

The switch element allows an author to specify a set of alternative elements from which only one acceptable element should be chosen. An element is acceptable if the element is a SMIL 1.0 element, the media-type can be decoded, and all of the test-attributes (see [Section 4.4](#)) of the element evaluate to "true".

An element is selected as follows: the player evaluates the elements in the order in which they occur in the switch element. The first acceptable element is selected at the exclusion of all other elements within the switch.

Thus, authors should order the alternatives from the most desirable to the least desirable. Furthermore, authors should place a relatively fail-safe alternative as the last item in the <switch> so that at least one item within the switch is chosen (unless this is explicitly not desired). Implementations should NOT arbitrarily pick an object within a <switch> when test-attributes for all fail.

Note that http URIs provide for content-negotiation, which may be an alternative to using the "switch" element in some cases.

Attributes

The switch element can have the following attributes:

id	Defined in Section 2
title	Defined in Section 3.3.1 It is strongly recommended that all switch elements have a "title" attribute with a meaningful description Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.

Element Content

If the "switch" element is used as a direct or indirect child of a "body" element, it can contain the following children:

a	<div>file:///Main%202Gb/Textures/ LECTURE%20STUFF/Multimedia/SMIL/</div>
---	--

- animation
 - Defined in [Section 4.5.1](#)
- audio
 - Defined in [Section 4.2.3](#)
- img
 - Defined in [Section 4.2.3](#)
- par
 - Defined in [Section 4.2.1](#)
- ref
 - Defined in [Section 4.2.3](#)
- seq
 - Defined in [Section 4.2.2](#)
- switch
 - Defined in [Section 4.3](#)
- text
 - Defined in [Section 4.2.3](#)
- textstream
 - Defined in [Section 4.2.3](#)
- video
 - Defined in [Section 4.2.3](#)

All of these elements may appear multiple times as children of a "switch" element.

If the "switch" element is used within a "head" element, it can contain the following child:

- layout
 - Defined in [Section 3.2](#)
 - Multiple layout elements may occur within the switch element.

4.4 Test Attributes

This specification defines a list of test attributes that can be added to any synchronization element, and that test system capabilities and settings. Conceptually, these attributes represent boolean tests. When one of the test attributes specified for an element evaluates to "false", the element carrying this attribute is ignored.

Within the list below, the concept of "user preference" may show up. User preferences are usually set by the playback engine using a preferences dialog box, but this specification does not place any restrictions on how such preferences are communicated from the user to the SMIL player.

The following test attributes are defined in SMIL 1.0:

- system-bitrate
 - This attribute specifies the approximate bandwidth, in bits per second available to the system. The measurement of bandwidth is application specific, meaning that applications may use sophisticated measurement of end-to-end connectivity, or a simple static setting controlled by the user. In the latter case, this could for instance be used to make a choice based on the users connection to the network. Typical values for modem users would be 14400, 28800, 56000 bit/s etc. Evaluates to "true" if the available system bitrate is equal to or greater than the given value. Evaluates to "false" if the available system bitrate is less than the given value.
 - The attribute can assume any integer value greater than 0. If the value exceeds an implementation-defined maximum bandwidth value, the attribute always evaluates to "false".
- system-captions
 - This attribute allows authors to distinguish between a redundant text equivalent of the audio portion of the presentation (intended for a audiences such as those with hearing disabilities or those learning to read who want or need this information) and text intended for a wide audience. The attribute can has the value "on" if the user has indicated a desire to see closed-captioning information, and it has the value "off" if the user has indicated that they don't wish to see such information. Evaluates to "true" if the value is "on", and evaluates to "false" if the value is "off".
- system-language
 - The attribute value is a comma-separated list of language names as defined in [RFC1766].

Evaluates to "true" if one of the languages indicated by user preferences exactly equals one of the languages given in the value of this parameter, or if one of the languages indicated by

user preferences exactly equals a prefix of one of the languages given in the value of this parameter such that the first tag character following the prefix is "-".

Evaluates to "false" otherwise.

Note: This use of a prefix matching rule does not imply that language tags are assigned to languages in such a way that it is always true that if a user understands a language with a certain tag, then this user will also understand all languages with tags for which this tag is a prefix.

The prefix rule simply allows the use of prefix tags if this is the case.

Implementation note: When making the choice of linguistic preference available to the user, implementors should take into account the fact that users are not familiar with the details of language matching as described above, and should provide appropriate guidance. As an example, users may assume that on selecting "en-gb", they will be served any kind of English document if British English is not available. The user interface for setting user preferences should guide the user to add "en" to get the best matching behavior.

Multiple languages MAY be listed for content that is intended for multiple audiences. For example, a rendition of the "Treaty of Waitangi", presented simultaneously in the original Maori and English versions, would call for:

```
<audio src="foo.rm" system-language="mi, en"/>
```

However, just because multiple languages are present within the object on which the system-language test attribute is placed, this does not mean that it is intended for multiple linguistic audiences. An example would be a beginner's language primer, such as "A First Lesson in Latin," which is clearly intended to be used by an English-literate audience. In this case, the system-language test attribute should only include "en".

Authoring note: Authors should realize that if several alternative language objects are enclosed in a "switch", and none of them matches, this may lead to situations such as a video being shown without any audio track. It is thus recommended to include a "catch-all" choice at the end of such a switch which is acceptable in all cases.

- system-overdub-or-caption
 - This attribute is a setting which determines if users prefer overdubbing or captioning when the option is available. The attribute can have the values "caption" and "overdub". Evaluates to "true" if the user preference matches this attribute value. Evaluates to "false" if they do not match.
- system-required
 - This attribute specifies the name of an extension. Evaluates to "true" if the extension is supported by the implementation, otherwise, this evaluates to "false". In a future version of SMIL, this attribute value will be an XML namespace [NAMESPACES].
- system-screen-size
 - Attribute values have the following syntax:
screen-size-val ::= screen-height"X"screen-width
Each of these is a pixel value, and must be an integer value greater than 0. Evaluates to "true" if the SMIL playback engine is capable of displaying a presentation of the given size. Evaluates to "false" if the SMIL playback engine is only capable of displaying a smaller presentation.
- system-screen-depth
 - This attribute specifies the depth of the screen color palette in bits required for displaying the element. The value must be greater than 0. Typical values are 1, 8, 24 Evaluates to "true" if the SMIL playback engine is capable of displaying images or video with the given color depth. Evaluates to "false" if the SMIL playback engine is only capable of displaying images or video with a smaller color depth.

4.5 Hyperlinking Elements

The link elements allows the description of navigational links between objects.

SMIL provides only for in-line link elements. Links are limited to uni-directional single-headed links (i.e. all links have exactly one source and one destination resource). All links in SMIL are actuated by the user.

Handling of Links in Embedded Documents

Due to its integrating nature, the presentation of a SMIL document may involve other (non-SMIL) applications or plug-ins. For example, a SMIL browser may use an HTML plug-in to display an embedded HTML page. Vice versa, an HTML browser may use a SMIL plug-in to display a SMIL document embedded in an HTML page.

In such presentations, links may be defined by documents at different levels and conflicts may arise. In this case, the link defined by the containing document should take precedence over the link defined by the embedded object. Note that since this might require communication between the browser and the plug-in, SMIL implementations may choose not to comply with this recommendation.

If a link is defined in an embedded SMIL document, traversal of the link affects only the embedded SMIL document.

If a link is defined in a non-SMIL document which is embedded in a SMIL document, link traversal can only affect the presentation of the embedded document and not the presentation of the containing SMIL document. This restriction may be released in future versions of SMIL.

Addressing

SMIL supports name fragment identifiers and the '#' connector. This means that SMIL supports locators as currently used in HTML (e.g. it uses locators of the form "http://foo.com/some/path#anchor1").

Linking to SMIL Fragments

A locator that points to a SMIL document may contain a fragment part (e.g. http://www.w3.org/test.smi#par1). The fragment part is an id value that identifies one of the elements within the referenced SMIL document. If a link containing a fragment part is followed, the presentation should start as if the user had fast-forwarded the presentation represented by the destination document to the effective begin of the element designated by the fragment.

The following special cases can occur:

- 1. The element addressed by the link has a "repeat" attribute.
 - 1. If the value of the "repeat" attribute is N, all N repetitions of the element are played.
 - 2. If the value of the "repeat" attribute is "indefinite", playback ends according to the rules defined for repeat value "indefinite".
- 2. The element addressed by the link is contained within another element that contains a "repeat" attribute.
 - 1. If the value of the "repeat" attribute is N, playback starts at the beginning of the element addressed by the link, followed by N-1 repetitions of the element containing the "repeat" attribute.
 - 2. If the value of the "repeat" attribute is "indefinite", playback starts at the beginning of the element addressed by the link. Playback ends according to the rules defined for repeat value "indefinite".
- 3. The element addressed by the link is content of a "switch" element: It is forbidden to link to elements that are the content of "switch" elements.

4.5.1 The a Element

The functionality of the "a" element is very similar to the functionality of the "a" element in HTML 4.0 [HTML40]. SMIL adds an attribute "show" that controls the temporal behavior of the source when the link is followed. For synchronization purposes, the "a" element is transparent, i.e. it does not influence the synchronization of its child elements. "a" elements may not be nested. An "a" element must have an href attribute.

Attributes

The "a" element can have the following attributes:

id	Defined in Section 2
href	This attribute contains the URI of the link's destination. The "href" attribute is required for "a" elements.
show	This attribute controls the behavior of the source document containing the link when the link is followed. It can have one of the following values: <ul style="list-style-type: none">o "replace": The current presentation is paused at its current state and is replaced by the destination

- resource. If the player offers a history mechanism, the source presentation resumes from the state in which it was paused when the user returns to it.
- o "new": The presentation of the destination resource starts in a new context, not affecting the source resource.
- o "pause": The source presentation is paused at its current state, and the destination resource starts in a new context. When the display of the destination resource ends, the source presentation resumes from the state in which it was paused.

	The default value of "show" is "replace".
title	Defined in Section 3.3.1 . It is strongly recommended that all "a" elements have a "title" attribute with a meaningful description. Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.

Element Content

The "a" element can contain the following children:

animation	Defined in Section 4.2.3
audio	Defined in Section 4.2.3
img	Defined in Section 4.2.3
par	Defined in Section 4.2.1
ref	Defined in Section 4.2.3
seq	Defined in Section 4.2.2
switch	Defined in Section 4.3
text	Defined in Section 4.2.3
textstream	Defined in Section 4.2.3
video	Defined in Section 4.2.3

4.5.2 The anchor Element

The functionality of the "a" element is restricted in that it only allows associating a link with a complete media object. HTML image maps have demonstrated that it is useful to associate links with spatial subparts of an object. The anchor element realizes similar functionality for SMIL:

- 1. The anchor element allows associating a link destination to spatial and temporal subparts of a media object, using the "href" attribute (in contrast, the "a" element only allows associating a link with a complete media object).
- 2. The anchor element allows making a subpart of the media object the destination of a link, using the "id" attribute.
- 3. The anchor element allows breaking up an object into spatial subparts, using the "coords" attribute.
- 4. The anchor element allows breaking up an object into temporal subparts, using the "begin" and "end" attributes. The values of the begin and end attributes are relative to the beginning of the media object.

Attributes

The anchor element can have the following attributes:

begin	Defined in Section 4.2.1
coords	The value of this attribute specifies a rectangle within the display area of a visual media object. Syntax and semantics of this attribute are similar to the coords attribute in HTML image maps, when the link is

associated with a rectangular shape. The rectangle is specified by four length values: The first two values specify the coordinates of the upper left corner of the rectangle. The second two values specify the coordinates of the lower right corner of the rectangle. Coordinates are relative to the top left corner of the visual media object (see Figure 4.5). If a coordinate is specified as a percentage value, it is relative to the total width or height of the media object display area.

An attribute with an erroneous coords value is ignored (right-x smaller or equal to left-x, bottom-y smaller or equal to top-y). If the rectangle defined by the coords attribute exceeds the area covered by the media object, exceeding height and width are clipped at the borders of the media object.

Values of the coords attribute have the following syntax:

```
coords-value ::= left-x "," top-y "," right-x "," bottom-y
```

- end
 - Defined in [Section 4.2.1](#)
- id
 - Defined in [Section 2](#)
- show
 - Defined in [Section 4.5.1](#)
- skip-content
 - Defined in [Section 3.3.1](#)
- title
 - Defined in [Section 3.3.1](#)
 - It is strongly recommended that all anchor elements have a "title" attribute with a meaningful description. Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.