

Media Integration & Presentation

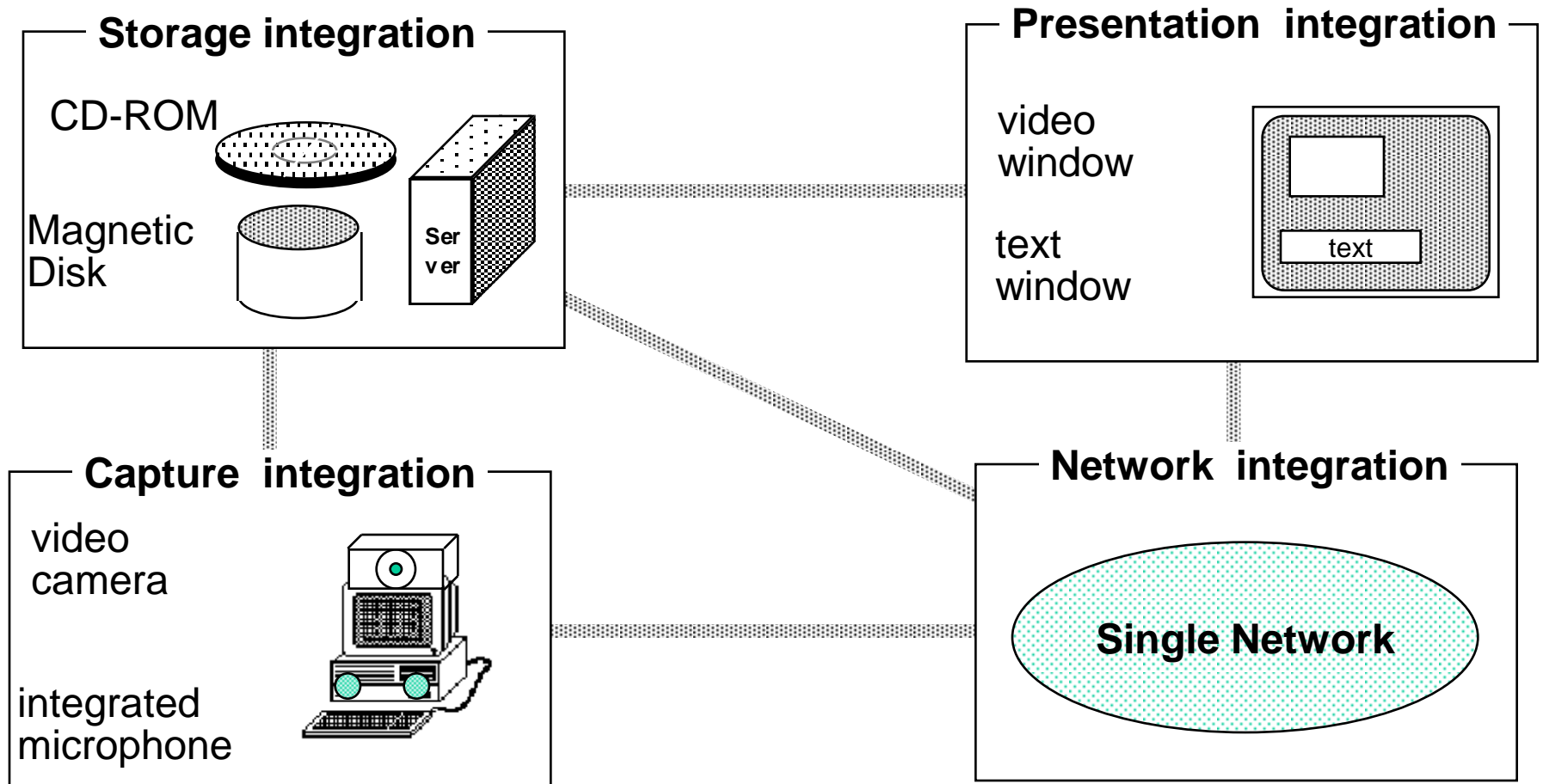
- Languages and Tools

- Media Integration Concept
- Media Synchronization and QoS
- Media Integration in Multimedia Presentation
- Media Integration Languages
- Media Integration Authoring Tools
- SMIL (Synchronous Multimedia Integration Language)
- HTML+TIME (Timed Interactive Multimedia Extension)
- VRML (Virtual Reality Modeling Language)

Media Integration Concept & Catalog

❑ Media integration

- Integrate different media into a system/application/file



Media Integration Concept & Catalog

❑ Media integration

- Core issues due to shared resources: CPU, memory, network, etc.

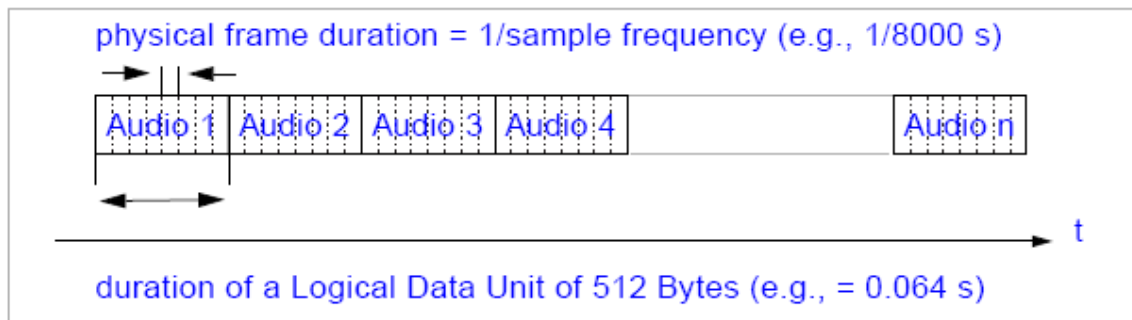
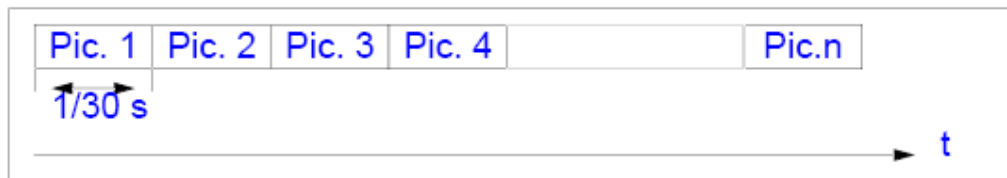
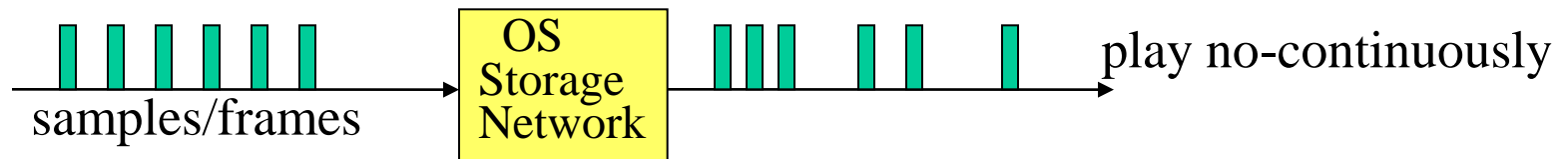
❑ Media integration catalog

- Media integration in operate system
- Media integration in storage system
- Media integration in database system
- Media integration in network system
- Media integration in human computer interface
- Media integration in message exchange
- Media integration in document representation
- Media integration in content presentation
-

➔ A special & important issue: media synchronization

Temporal Relations in Video and Audio

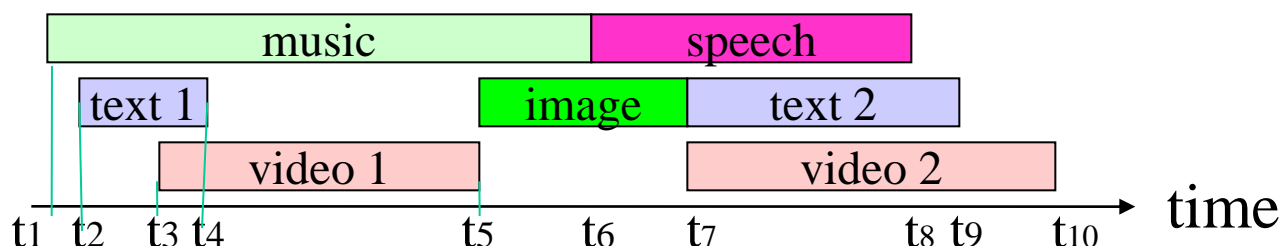
- ❑ Media are classified into
 - Discrete media (DM): text, still image, graphics image
 - Continuous media (CM): audio, video, animation
- ❑ CM are extremely time-sensitive !!!



Media Synchronization and QoS

Media synchronization: keep temporal relationships

- Intra-medium synchronization
- Inter-media synchronization



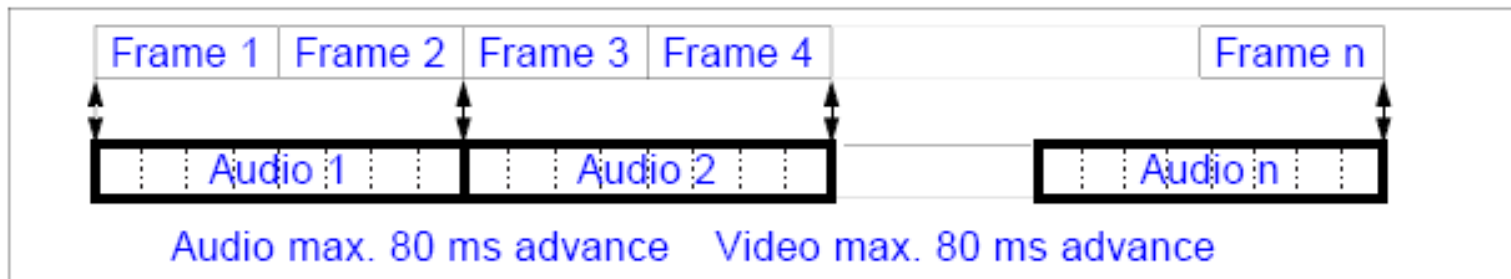
QoS (Quality of Service):

- Specify media quality
 - The set of parameters that define the properties of media objects/streams
 - Performance, error rate, delay, jitter, time skew, ...
 - How to guarantee QoS
- key technology in mm OS, storage, network, ...

Media		Mode, Application	QoS
Video	Animation	correlated	+/- 120 ms
	Audio	lip synchronization	+/- 80 ms
	Image	overlay	+/- 240 ms
		non-overlay	+/-500 ms
	Text	overlay	+/- 240 ms
		non-overlay	+/-500 ms
Audio	Animation	event correlation (e.g., dancing)	+/- 80 ms
	Audio	tightly coupled (stereo)	+/- 11 μ s
		loosely coupled (dialogue mode with various participants)	+/- 120 ms
		loosely coupled (e.g., background music)	+/- 500 ms
	Image	tightly coupled (e.g., music with notes)	+/- 5 ms
		loosely coupled (e.g., slide show)	+/- 500 ms
	Text	Anmerkungen zu Text	+/- 240 ms
	Pointer	Audio Related to the Item	-500ms +750 ms

Lip Synchronization

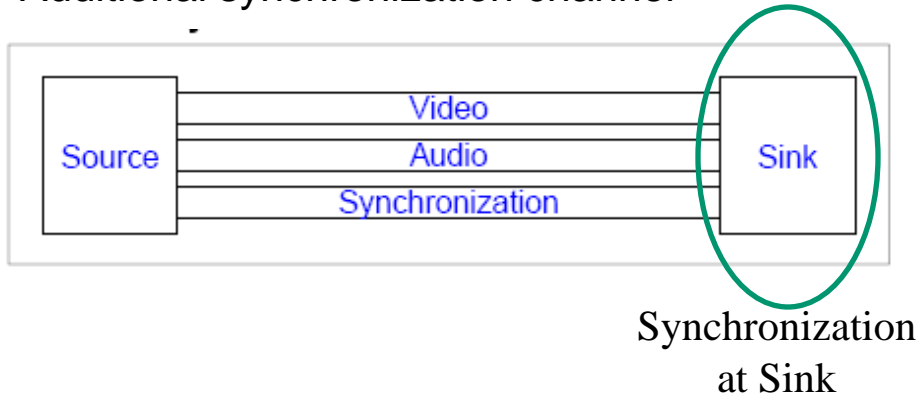
- Lip synchronization: Coupling between audio and video
- Acceptable Skew between video and audio: $\sim 100\text{ms}$



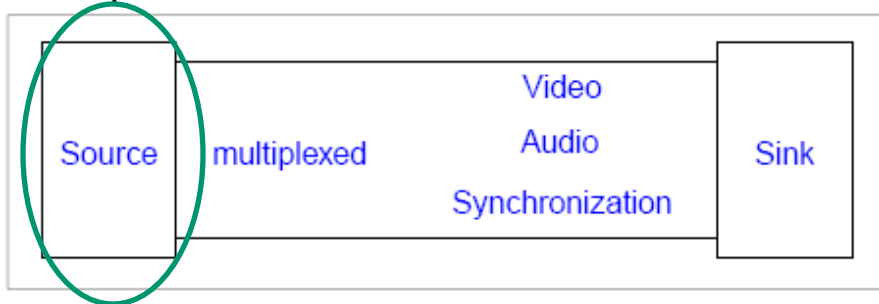
Synchronization Specification and Location

Issues: where to put synchronization data?
where to do synchronization task?

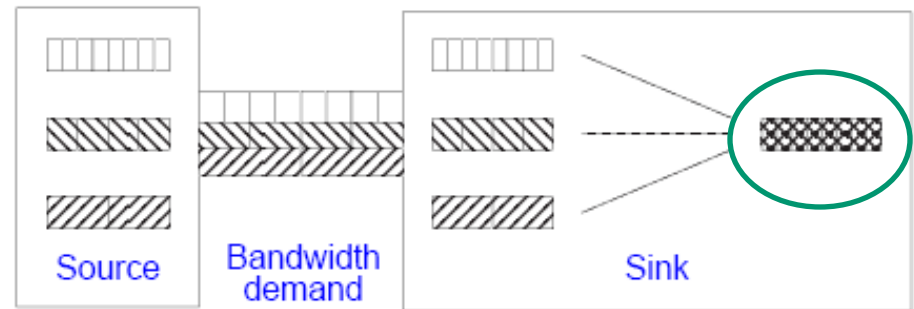
Additional synchronization channel



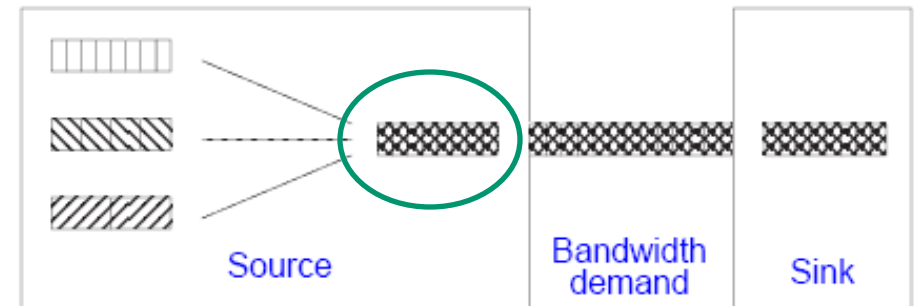
Multiplexed channel



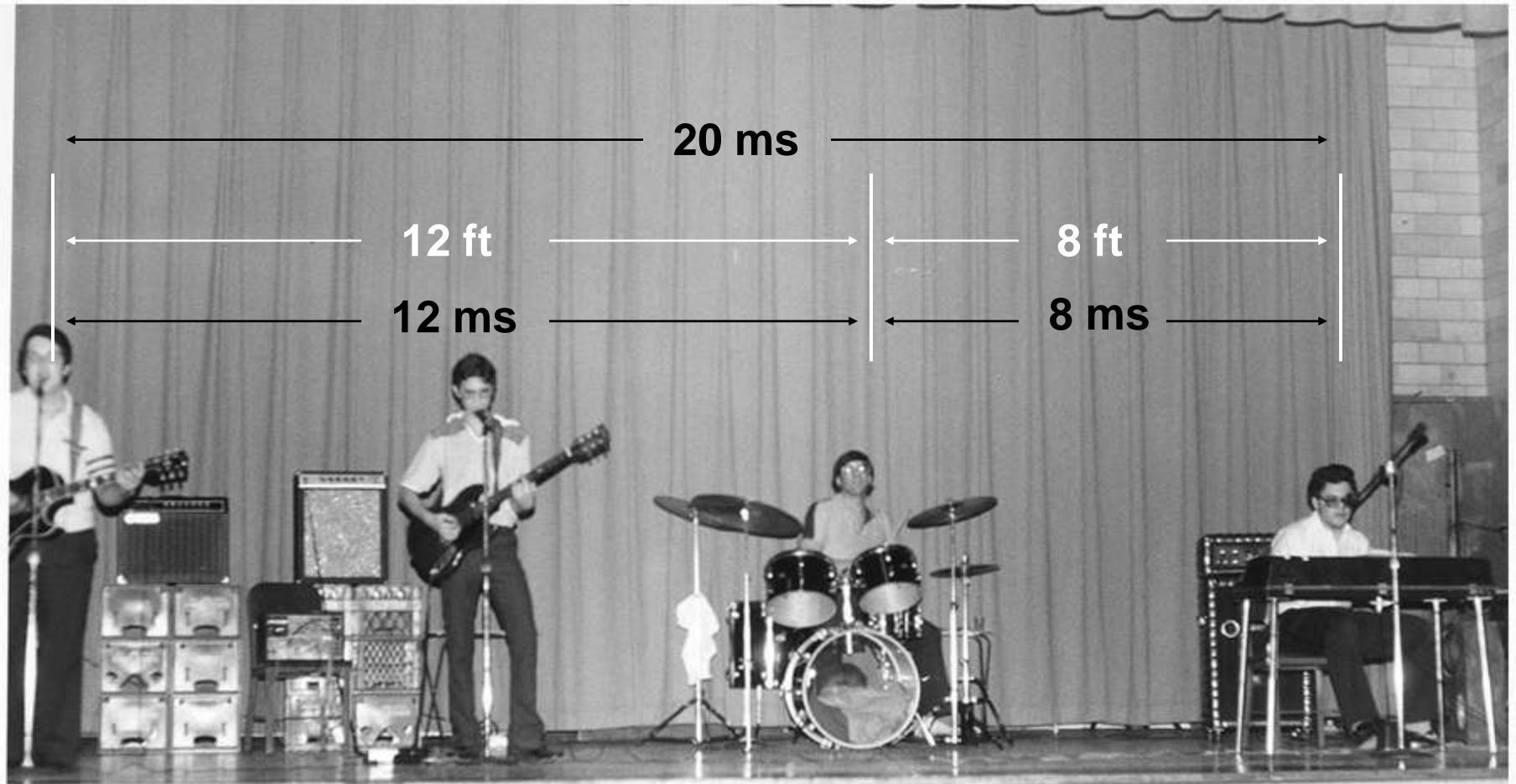
Synchronization at the sink node



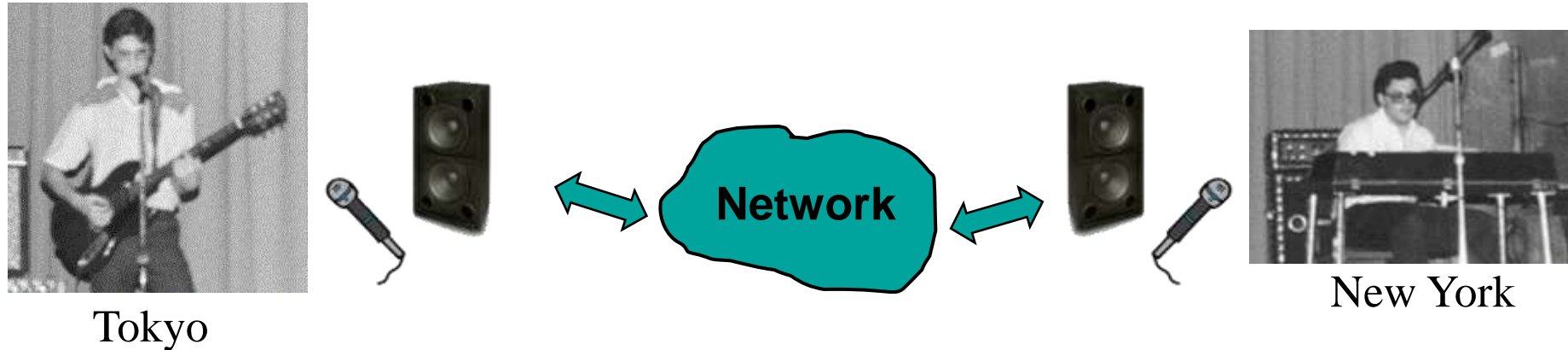
Synchronization at the source node



Music Performance



Distributed Music Over Network



- Adapt VOIP architecture for low latency:
 - Hosts use Real Time Protocol (RTP) to exchange audio streams
 - Effective if
 - host audio chain customized for low latency
 - low latency, over-provisioned network
 - Quality of Service (QoS) mechanisms (perhaps)
- Non-ideal network (BW limits, congestion, etc.)
 - Occasional packet delays and losses inevitable
 - Garbled sound (clicks and gurgles) due to small buffers

Media Integration in Multimedia Presentation

❖ Multimedia presentation

- A process to assembly and synchronize all media objects/components that you have prepared to create a final multimedia product, such as a electronic file, a slide show, a web document, an e-book, etc.

❖ Presentation control elements

- Where? Spatial attribute (location, size, ...)
- When ? Time attribute (start and end time, synchronization, ...)
- How ? Effective attribute (volume, transition, relationships, ...)

❖ Presentation creation methods

- via computer languages
- via authoring tools

Languages for Multimedia Presentation

General speaking, most of the computer languages are capable to make multimedia presentation products. But the following are often used:

Programming Language

- C/C++, Visual Basic
- Perl
- Java

Script Language

- JavaScript, ECMAScript (*European Computer Manufacturer's Association*)
- PHP
- Flash/Lingo (*Macromedia*)

Markup Language

- HTML, DHTML, XML, SVG (*Scalable Vector Graphics*)
- SMIL, HTML+TIME
- WML

Scene Description Language

- VRML
- BIFS (*Binary Format for Scene in MPEG-4*)
- DDL (*Description Definition Language in MPEG-7*)

Multimedia Authoring Tools

- ❑ **Using computer language** to make multimedia presentations
 - *Need programming skill, hard for beginner, small size and flexible function*
- ❑ **Using Authoring Tools** to make multimedia presentations
 - A visualized authoring window using drag-and-drop via mouse
 - Less or no need for programming skill, large size and limited functions
- ❑ **Multimedia authoring tools**
 - Slide show based: from slide to slide in sequence of forward or backward
 - PowerPoint, Kai's Power Show, QuickTime Pro
 - Digital movie based: from begin to end
 - Macromedia Director
 - Branch based: providing users with a choice over where to go
 - Macromedia Authorware, Clickteam's Multimedia Fusion, Asymetrix's Toolbox
 - Web file based: creating a web document
 - Macromedia Dreamweaver, Adobe Golive, Frontpage, Netscape Composer
 - VR/3D file based: creating a animation or wml file
 - Macromedia Flush, MS Liquid Motion

W3C Consortium

- ❑ **W3C**, founded in October 1994: <http://www.w3.org/>
- ❑ Purpose: develop common protocols that promote WWW's evolution and ensure its interoperability
- ❑ User Interface Domain
 - Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), Document Object Model (DOM), **SMIL**, **SVG**
- ❑ Technology and Society Domain
 - Platform for Internet Content Selection (PICS), Resource Description Framework (RDF), Platform for Privacy Preferences (P3P)
- ❑ Architecture Domain
 - Hypertext Transfer Protocol (HTTP), Extensible Markup Language (XML)

...Follow links at <http://www.w3.org/> for more details...

Embedding Audio/video into HTML

- ❑ Embed AV into a web page via programming/script language
 - Java applet
 - JavaScript or other scripts
- ❑ Embed AV into a web page via `<embed>` tag

➤ Embed audio

```
<embed src="path/MyAudio.wav" autostart="true" loop="true"></embed>
```

Note 1: The sound file begins to play as soon as it is loaded

Note 2: loop = "true" → play forever

Note 2: Plug-in is needed for playing audio file in .mov, .ra, .mp3, .aiff, etc.

➤ Embed video

```
<embed src="path/MyVideo.avi" width="320" height="240"  
      autostart="true" loop="true">
```

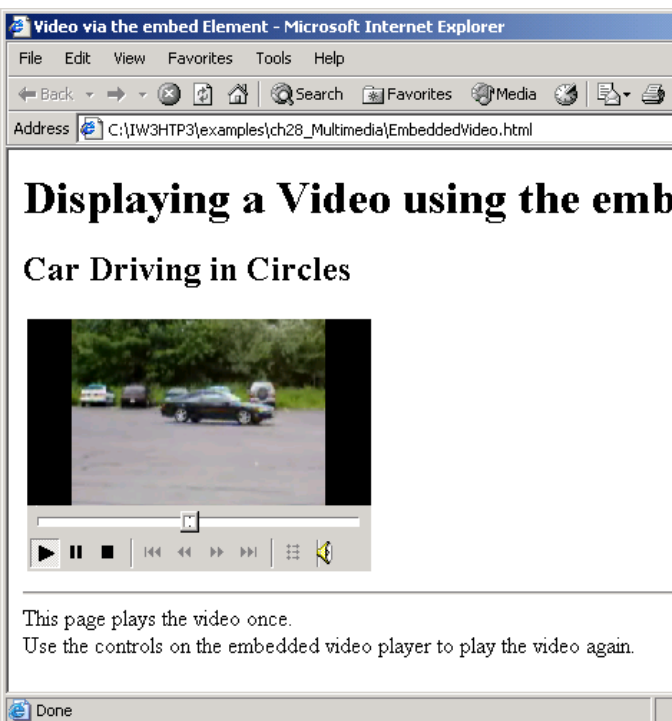
```
</embed>
```

Note 1: The sound file begins to play as soon as it is loaded

Note 2: Plug-in is needed for playing video file in .mov, .ra, .mpg, etc.

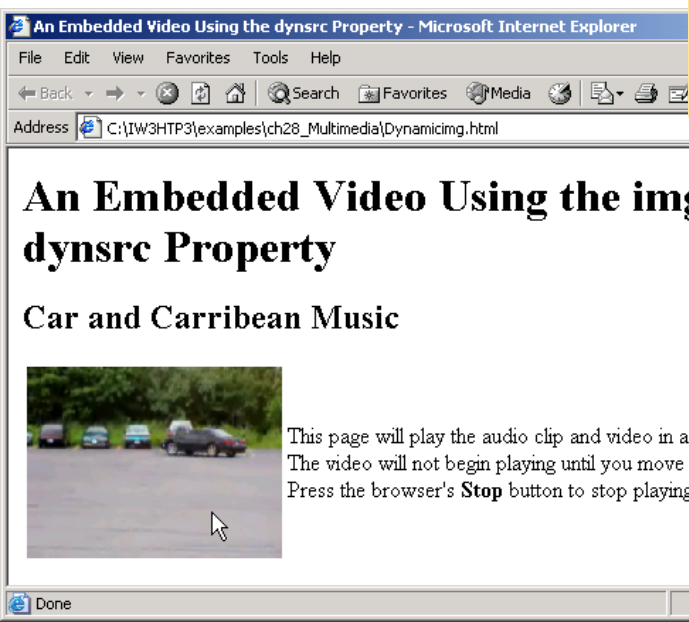
➤ Embed tag is not enough to play multiple synchronized media object

Add Audio/video onto Webpage Using embed tag



```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <!-- Fig. 28.4: EmbeddedVideo.html -->
6 <!-- video via the embed Element -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10     <title>Video via the embed Element</title>
11   </head>
12
13   <body>
14     <h1>Displaying a video using the embed Element</h1>
15     <h2>Car Driving in Circles</h2>
16
17     <table>
18       <tr><td><embed src = "car_hi.wmv" loop = "false"
19         width = "240" height = "176">
20       </embed></td>
21     </tr></table>
22
23     <hr />
24     This page plays the video once.<br />
25     Use the controls on the embedded video player to play the
26     video again.
27   </body>
28 </html>
```

Add Audio/video onto Webpage Using img & dynsrc



```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <!-- Fig. 28.2: Dynamicimg.html -->
6 <!-- Demonstrating the img element's dynsrc property -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10     <title>An Embedded Video Using the dynsrc Property</title>
11     <bgsound src =
12       "http://msdn.microsoft.com/downloads/sounds/carib.MID"
13       loop = "-1"></bgsound>
14   </head>
15
16   <body>
17     <h1>An Embedded Video Using the img element's
18       dynsrc Property</h1>
19     <h2>Car and Carribean Music</h2>
20     <table>
21       <tr><td><img dynsrc = "car_hi.wmv"
22         start = "mouseover" width = "180"
23         height = "135" loop = "-1"
24         alt = "Car driving in circles" /></td>
25       <td>This page will play the audio clip and video
26         in a loop.<br />The video will not begin
27         playing until you move the mouse over the
28         video.<br />Press the browser's<strong>Stop</strong>
29         button to stop playing the sound and the video.</td>
30     </tr>
31   </table>
32 </body>
33 </html>
```


SMIL- Synchronized Multimedia Integration Language

- ❑ Define an XML-based language that allows authors to write interactive multimedia presentations → describe the temporal behaviour of a multimedia presentation, associate hyperlinks with media objects and describe the layout of the presentation on a screen.
- ❑ XML application enabling author to specify what should be presented **when**
- ❑ SMIL 1.0 specification, June 1998
- ❑ SMIL 2.0, August 2001, SMIL 2.1, December 2005
- ❑ SMIL 3.0, December 2008
 - Define a set of reusable markup modules that define the semantics
 - # Animation module
 - # Content control module
 - # Transition effect module
 - #
 - Module reuse in other XML based languages: WML, SVG, MPEG-4, etc
 - Others

SMIL : Design Principles

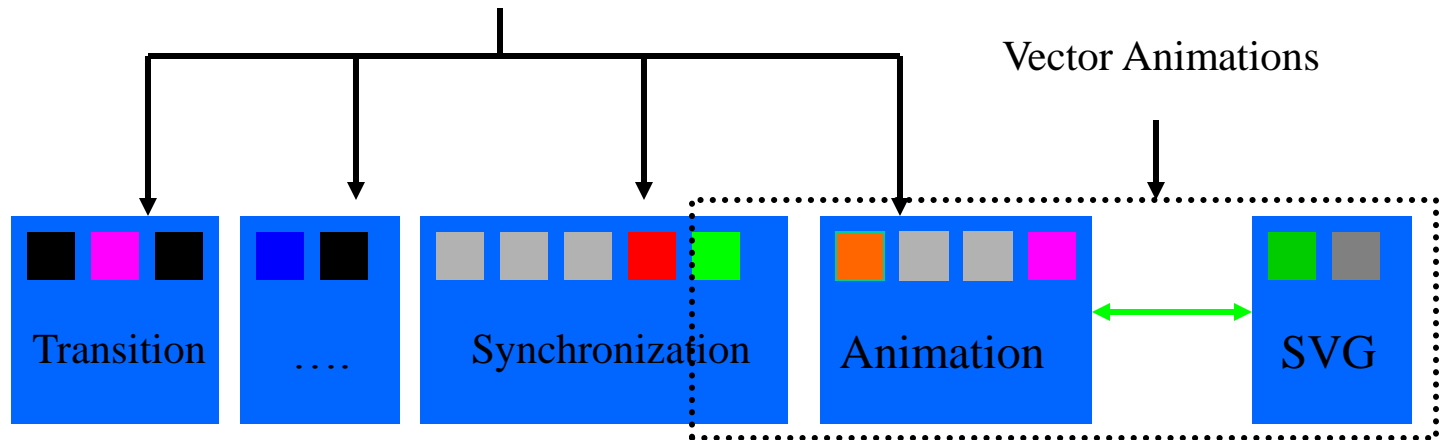
Meta-language which allows the description of multimedia documents ranging from the simplest to the very complex.

Languages space

1 application *profile*

Vector Animations

Functional space



Syntactic and
compositional
space,
programming
APIs, ...

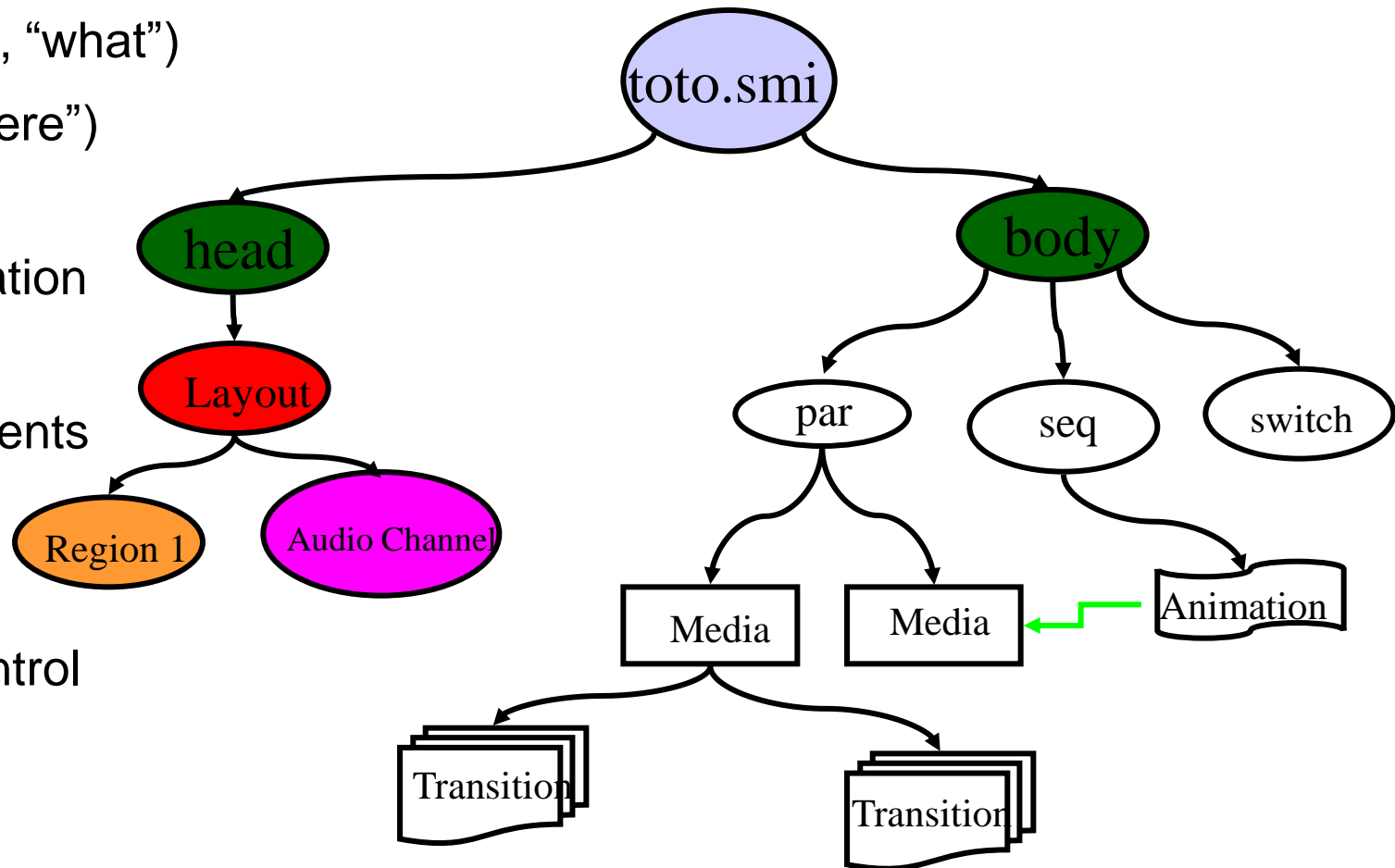
DOM 1-2
SMIL DOM

XML

Namespaces

SMIL Structure and Modules

- Structure
- Meta (“who”, “what”)
- Layout (“where”)
- Timing and Synchronization (“when”)
- Media Elements (“what”)
- Linking
- Content Control



SMIL High Level Document Structure

```
<smil>  
  <head>  
    <meta>  
      <!-- ... information about the document ... -->  
    </meta>  
    <layout>  
      <!-- ... definitions used for the spatial layout ... -->  
    </layout>  
  </head>  
  <body>  
    <!-- ... objects, temporal relations, links ... -->  
  </body>  
</smil>
```

SMIL Meta

```
<smil>  
  <head>  
    <meta ... />  
  </head>  
</smil>
```

The meta elements contain information describing the document, either to inform the human user or to assist some automation, e.g.,

```
<meta name="title" content="My Italy Trip"/>  
<meta name="copyright" content="©1998 WGBH" />  
<meta name="base" content="http://billswin.edu/Italy/" />
```

SMIL Layout

```
<smil>
<head>
  <layout>
    <root-layout ... />
    <region id="R1" ... />
    <region id="R2" ... />
  </layout>
</head>
</smil>
```

Includes the <layout>, <root-layout>, <region> elements, and related attributes.

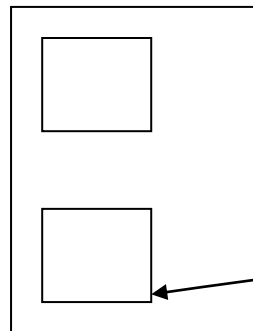


Example

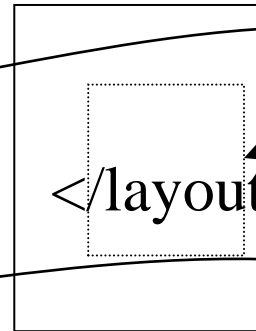
```
<root-layout height="300" width="450"
  background-color="#FFFFFF"
  title="Venezia!"/>
<root-layout height="450" width="625"
  background-color="black"/>

<region id="title" left="5" top="150"
  width="400" height="200"
  z-index="1"/>
<region id="videoregion" top="0"
  left="0" height="240"
  width="352"/>
```

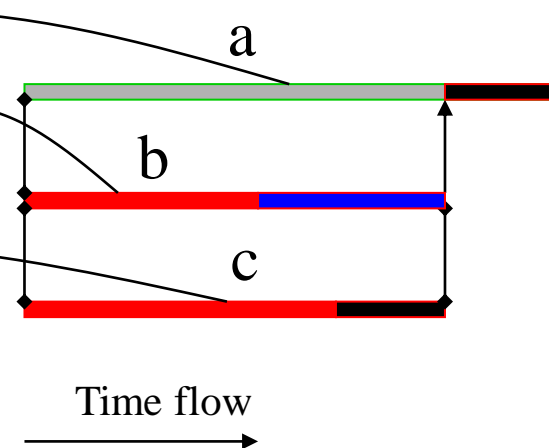
Region 1



Region 2



Region 3



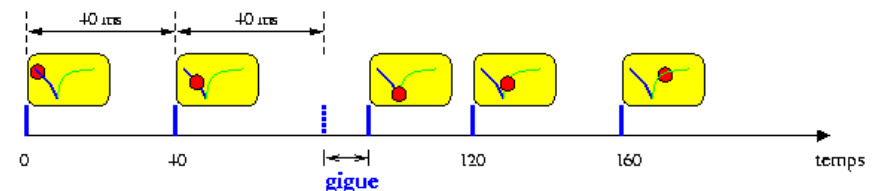
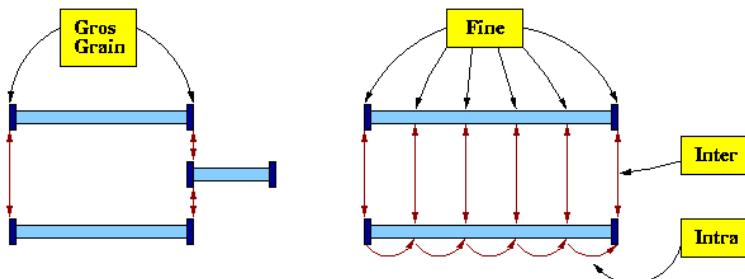
SMIL Timing and Synchronization

```
<smil>
  <body>
    <!-- ... timing included here ... -->
  </body>
</smil>
```

- Sequence and parallel timelines, via **<seq>** and **<par>**
- Timing control properties, via **“begin”**, **“end”**, **“repeat”**, **“dur”**
- “The accuracy of synchronization between the children in a parallel group is implementation-dependent”
 - “soft synchronization” vs. “hard synchronization”
 - how to realize hard synchronization

Hard vs. Soft Synchronization

- Hard synchronization: player synchronizes the children in the "par" (parallel play) element to a common clock
- Soft synchronization: each child of the "par" element has its own independent clock
- syncBehavior
 - **canSlip** : the synchro is loose, child elements can slip from the parent clock
 - **locked** : the Synchronization is hard (lipsync), amount of tolerated slipping (syncTolerance).
 - **Independent** : synchro completely independent
- syncTolerance = "amount of jitter"
- syncMaster = "true" clock ticker of the par element



SMIL Media Elements

<smil>

 <body>

 <!-- ... media elements included here ... -->

 </body>

</smil>

- Includes the media declaration elements <text>, , <audio>, <video>, <textstream>, <animation>, and <ref>
- all elements (animation, img, ref, text, textstream and video) are contained within a single containing block defined by the root-layout element

SMIL Linking

<smil>

<body>

<!-- ... linking included here ... -->

</body>

</smil>

- Includes the <a> and <anchor> elements, e.g.,

<text src="media/w3c.txt" region="w3c"
begin="14.05s" dur="15.95s" />

- Timing applied to HTML <a> and <area> tags could provide much or all of SMIL functionality; hence, linking modules under review

An Example of SMIL File

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
    <layout>
      <topLayout width="640px" height="480px">
        <region id="whole" top="0px" left="0px"
          width="640px" height="480px" />
      </topLayout>
    </layout>
  </head>
  <body>
    <seq>
      
      
        dur="3s"/>
      
      
    </seq>
  </body>
</smil>
```

SMIL Browsers and Authoring Tools

SMIL browser

- [RealOne Platform](#) by RealNetworks with full support for the SMIL 2.0
- [GRiNS for SMIL-2.0](#) by Oratrix provides a SMIL 2.0 player
- [Internet Explorer 6.0](#) by Microsoft including [XHTML+SMIL Profile](#)
- [X-Smiles](#), version 0.4 a new java-based XML browser

Authoring Tools

- [GRiNS Editor](#) by Oratrix based on SMIL2 Editor family and streamlined
- [SMILGen](#) by RealNetworks, a SMIL (and XML) authoring tool
- [Ezer](#) by SMIL Media
- [Fluition](#) by Confluent Technologies
- [Grins](#) by Oratrix

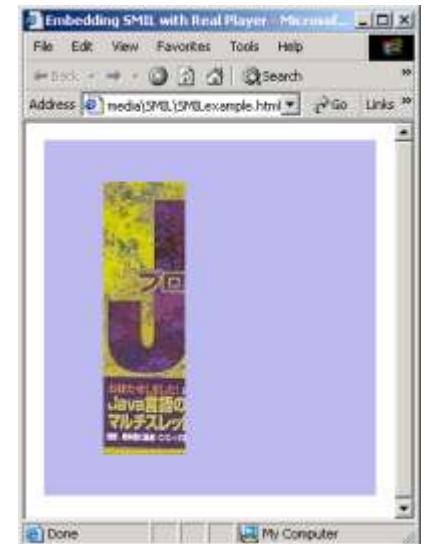
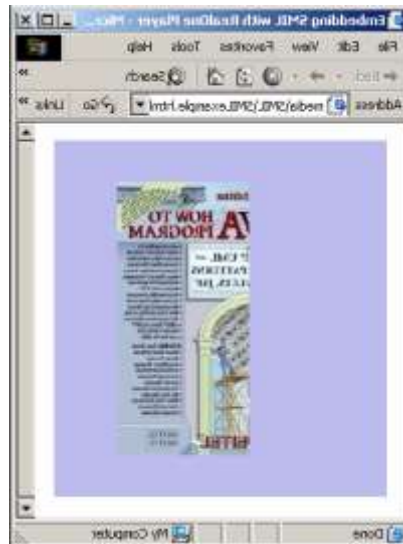
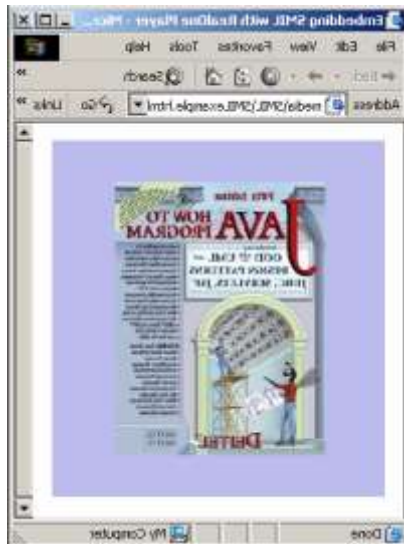
Another Example of SMIL File

```
1 <smil xmlns="http://www.w3.org/2000/SMIL20/CR/Language">
2
3   <!-- Fig. 20.15 : exampleSMIL.smil -->
4   <!-- Example SMIL Document      -->
5
6   <head>
7     <layout>
8       <root-layout height = "300" width = "280"
9         background-color = "#bbbbee" title = "Example" />
10
11       <region id = "image1" width = "177" height = "230"
12         top = "35" left = "50" background-color = "#bbbbee" />
13     </layout>
14
15     <transition id = "wipeForward" dur = "2s" type = "barwipe" />
16     <transition id = "wipeBackward" dur = "2s" type = "barwipe"
17       subtype = "topToBottom" />
18
19     <transition id = "fadeIn" dur = "2s" type = "fade"
20       subtype = "fadeFromColor" fadeColor = "#bbbbee" />
21
22     <transition id = "fadeOut" dur = "2s" type = "fade"
23       subtype = "fadeToColor" fadeColor = "#bbbbee" />
24
```

```
25   <transition id = "crossFade" type = "fade" subtype = "crossfade"
26     dur = "2s" />
27
28 </head>
29 <body>
30   <seq>
31     <par>
32       <img src = "book1.jpg" region = "image1"
33         transIn = "wipeForward" transOut = "wipeForward"
34         alt = "book1" dur = "6s" fill = "transition"
35         fit = "fill" />
36       <audio src = "bounce.au" dur = ".5s" />
37     </par>
38     <par>
39       <img src = "book2.jpg" region = "image1" transIn = "fadeIn"
40         transOut = "fadeOut" alt = "book2" dur = "6s"
41         fit = "fill" fill = "transition" />
42       <audio src = "bounce.au" dur = ".5s" />
43     </par>
44     <par>
45       <img src = "book3.jpg" region = "image1"
46         transIn = "wipeBackward" transOut = "fadeOut"
47         alt = "book3" dur = "6s" fit = "fill"
48         fill = "transition" />
49       <audio src = "bounce.au" dur = ".5s" />
50     </par>
51     <par>
52       <img src = "book4.jpg" region = "image1" transIn = "crossFade"
53         transOut = "fadeOut" alt = "book4" dur = "6s"
54         fit = "fill" fill = "transition" />
55       <audio src = "bounce.au" dur = ".5s" />
56     </par>
57     <par>
58       <img src = "book5.jpg" region = "image1"
59         transIn = "wipeForward" transOut = "wipeBackward"
60         alt = "book5" dur = "6s" fit = "fill"
61         fill = "transition" />
62       <audio src = "bounce.au" dur = ".5s" />
63     </par>
64     <par>
65       <img src = "book6.jpg" region = "image1"
66         transIn = "crossFade" alt = "book6" dur = "6s"
67         fit = "fill" fill = "transition" />
68       <audio src = "bounce.au" dur = ".5s" />
69     </par>
70   </seq>
71
72 </body>
73 </smil>
```

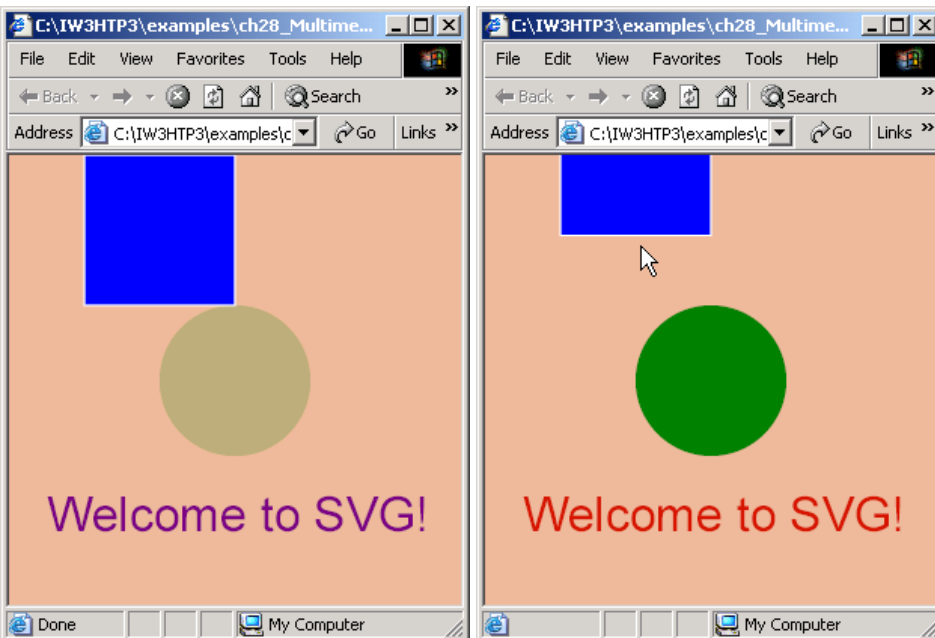
Another Example of SMIL File

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <!-- Fig. 28.16: SMILexample.html -->
6 <!-- embedding SMIL with RealOne Player -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10     <title>Embedding SMIL with RealOne Player</title>
11   </head>
12   <body>
13     <div style = "text-align: center">
14       <embed src = "exampleSMIL.smil"
15         controls = "Imagewindow"
16         type = "audio/x-pn-realaudio-plugin"
17         width = "280" height = "300" autostart = "true">
18     </embed>
19   </div>
20 </body>
21 </html>
```



SVG Scalable Vector Graphics

Produced by mathematical
equations via XML vocabulary



```
1 <?xml version = "1.0"?>
2
3 <!-- Fig. 28.17 : shapes.svg -->
4 <!-- Simple example of SVG -->
5
6 <svg viewBox = "0 0 300 300" width = "300" height = "300">
7
8   <!-- Generate a background -->
9   <g>
10      <path style = "fill: #eebb99" d = "M0,0 h300 v300 h-300 z"/>
11   </g>
12
13   <!-- Circle shape and attributes -->
14   <g>
15
16      <circle style = "fill:green;" cx = "150" cy = "150" r = "50">
17         <animate attributeName = "opacity" attributeType = "CSS"
18            from = "0" to = "1" dur = "6s" />
19      </circle>
20
21   <!-- Rectangle shape and attributes -->
22   <g>
23
24      <rect style = "fill: blue; stroke: white"
25         x = "50" y = "0" width = "100" height = "100">
26         <animate attributeName = "y" begin = "mouseover" dur = "2s"
27            values = "0; -50; 0; 20; 0; -10; 0; 5; 0; -3; 0; 1; 0" />
28      </rect>
29
30   <!-- Text value and attributes -->
31   <g>
32
33      <text style = "fill: red; font-size: 24pt"
34         x = "25" y = "250"> Welcome to SVG!
35      <animateColor attributeName = "fill"
36         attributeType = "CSS" values = "red;blue;yellow;green;red"
37         dur = "10s" repeatCount = "indefinite"/>
38   </text>
39 </g>
</svg>
```

HTML+TIME (Timed Interactive Multimedia Extensions)

- ❑ Proposed by Microsoft, presently not been endorsed by W3C
- ❑ HTML+TIME 1.0 is based on SMIL 1.0 and supported in IE5+
- ❑ HTML+TIME 2.0 is based on SMIL 2.0 and supported in IE 5.5+
- ❑ Add timing and media synchronization support to HTML pages
 - media elements: `t:ANIMATION`, `t:AUDIO`, `t:VIDEO`, `t:IMG`
 - control elements: `t:EXCL`, `t:SEQ`, `t:PAR`
- ❑ Use both timeline model and event-driven model
 - `BEGIN`, `DUR`, `BeginWith`
- ❑ HTML+TIME structure

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<STYLE> .time {behavior: url(#default#time2);} </STYLE>
<?IMPORT namespace="t" implementation="#default#time2">
</HEAD>
<BODY>

    . . . . .

</BODY>
</HTML>
```


VRML (Virtual Reality Modeling Language)

- ❑ Pronounced either V-R-M-L or “Vermal”
- ❑ A language that describes geometry and behavior of a 3D scene or “world”
- ❑ Based on SGI’s Moving World languages
- ❑ SMIL 1.0 (1995), VRML 2.0/VRML97, ISO standard (ISO/IEC-14772-1:1997)
- ❑ “World” can be single or a group of files, ranged from simple to complex scene
- ❑ A VRML file is a plain UTF-8 or ASCII text file ended with `.wrl`
- ❑ Use a plain text editor (e.g. Notepad) to input, modify and save a VRML file

```
#VRML V2.0 utf8
DEF APP Appearance {material Material{ diffuseColor 1 0 0 } }
Shape{ appearance USE APP
      geometry Cylinder{ radius 1 height 5 } }
```

- ❑ VRML file can be viewed by a specific VRML browser or a Web browser with plug-in
- 3D ObjectViewer, Cosmo Player, Community Place, GL View, WebDimension, WorldView, etc.
- ❑ VRML file can be embedded in a HTML file via `<Frame>`, `<EMBED>`, `<OBJECT>`
`<FRAME SRC="my.wrl" width="300" height="280">`
`<EMBED SRC="my.wrl" width="300" height="280">`

Demos

☐ SMIL

☐ SVG

☐ VRML