

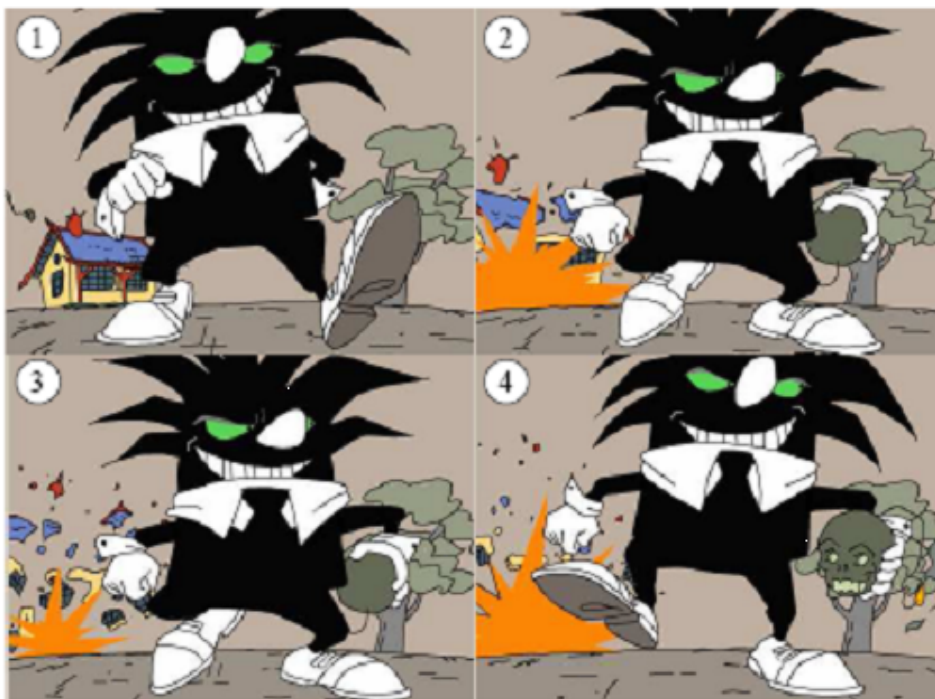
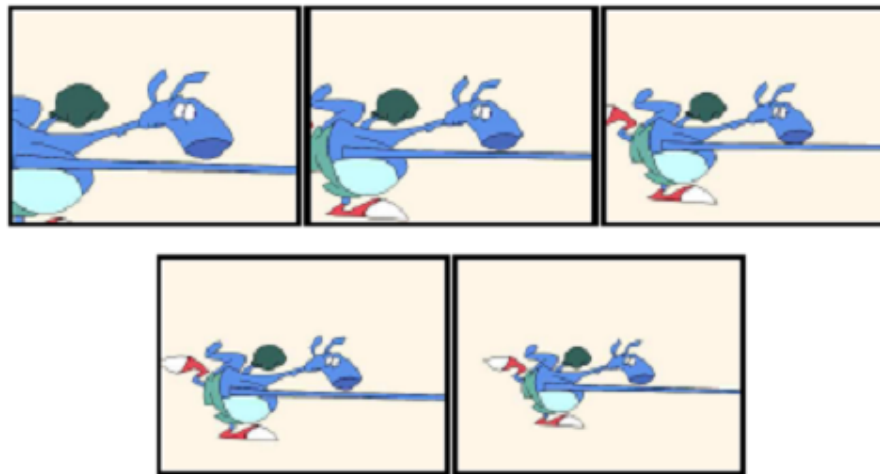
ANIMATING THE WEB

0

USE CASES

- User interface effects
 - Visual clues of possible user interactions
 - Transitions between states in the application
- Story telling
 - Cartoon animations
 - Ads
- Data Visualization
 - Scientific
 - Newspapers (e.g. voting polls)
 - Maps

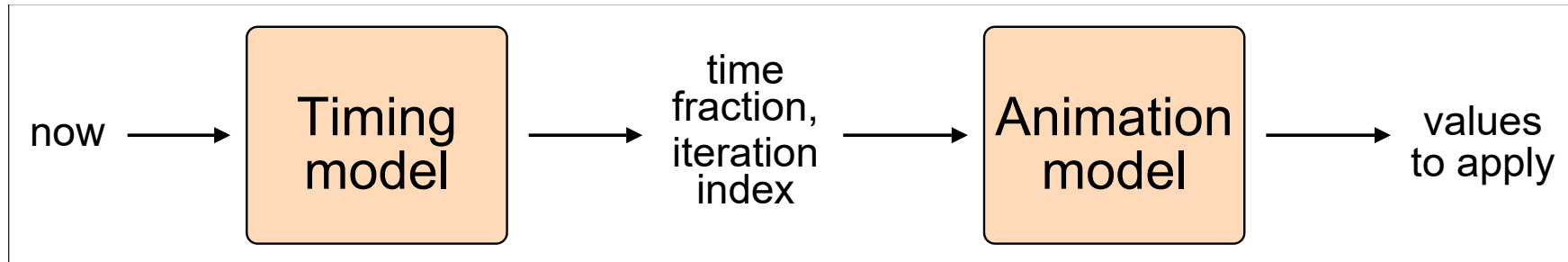
GRAPHICAL ANIMATION EXAMPLES



TECHNOLOGIES

- Started with D(ynamic) HTML
- Continued with SMIL
- Evolved into
 - JavaScript APIs
 - setTimeout, ...
 - Declarative solutions
 - SMIL/SVG Animations
 - CSS Animations
 - CSS Transitions
- Basic concepts
 - Timing model
 - Animation model

TIMING AND ANIMATION MODELS



SMIL OVERVIEW

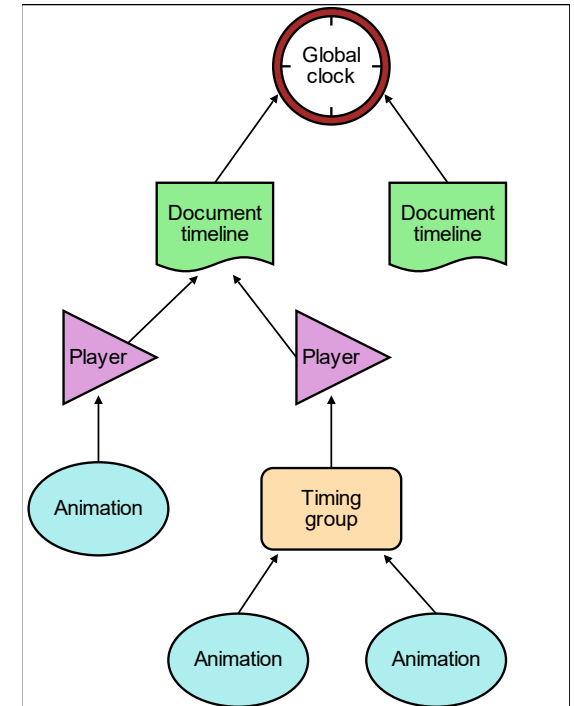
- Recommendation from W3C
 - SMIL 1.0 in 1998
 - SMIL 3.0 in 2008
- Time-oriented language divided into many modules
 - Simple-to-Advanced Timing: nested timing containers
 - Simple-to-Advanced Synchronization: timelines, locked, ...
 - Animations and transition effects
 - Media independent: audio, video, text ...
 - Simple 2D layout: regions
- Limited support in browsers / players

SMIL EXAMPLE

```
<smil>
  <head>
    <layout>
      <root-layout width="700" height="400" background-color="white" />
      <region id="region_2" left="0" top="10" width="100%" height="90%" />
      <region id="region_1" left="0" top="0" width="1" height="1" />
    </layout>
  </head>
  <body>
    <par>
      <audio id="aSound" src="a.mp3" region="region_1"/>
      <seq id="txt">
        <text id="txt1" src="texte/text1.rt" region="region_2" dur="9s" />
        <text id="txt2" src="texte/text2.rt" region="region_2" dur="7s" />
        <text src="texte/text3.rt" region="region_2" dur="17s" />
        ...
      </seq>
    </par>
  </body>
</smil>
```

TIMING HIERARCHY

- Timeline vs Document
 - 1 clock vs. Multiple clocks
 - Time container
- Timed Elements
 - Some elements in the DOM tree define timing behavior
 - Resetting the timeline
 - ex. SMIL seq/par



TYPICAL ANIMATION MODELS

Discrete animations

```
<smil>
  <body>
    <seq repeatCount="indefinite">
      
      
    </seq>
  </body>
</smil>
```

Animations by interpolation

```
<svg>
  <rect width="120" height="50" fill="blue" >
    <animate begin="10s" dur="2s" repeatCount="5" attributeName="fill"
      from="blue" to="red" fill="freeze" />
  </rect>
</svg>
```

SVG TIMING

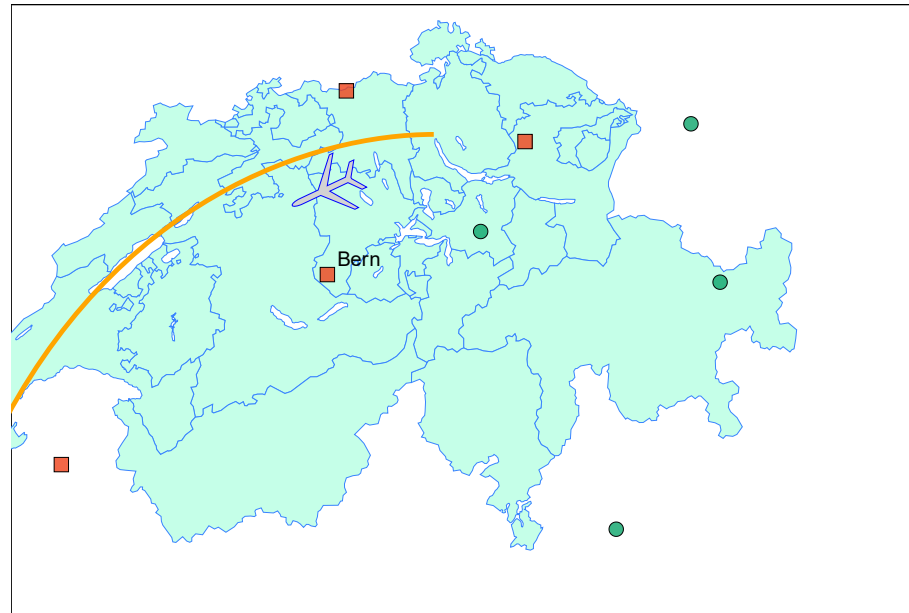
- SVG uses a restricted SMIL Timing
 - Timed elements: animation elements and media elements
 - Timed containers
 - SVG has only one time container (one timeline per document)
 - Media elements (in Tiny 1.2) define their own time lines, with simplified clock management
- Attributes to control the timing
 - Common to media (Tiny 1.2) and animations
 - Attributes begin, end, dur, repeatDur ...
- Attributes to control the synchronization (Tiny 1.2)
 - Attributes syncBehavior, syncTolerance

SVG ANIMATIONS

- SVG reuses SMIL animation as is
 - Simple and complex animations possible
 - Definition of the interface with CSS (Sandwich animation model)
- Elements for animations
 - `<animate>`, `<animateColor>`, `<animateMotion>`, `<animateTransform>`, `<set>`
- Attributes for animations
 - `xlink:href`, `attributeName`: to define the target attribute to animate
 - `from`, `to`, `values`: to give the interpolation values
 - `calcMode`, `keyTimes`, `keySplines`: to give the interpolation type
 - `additive`, `accumulate`: to give the behavior when multiple animations target the same attribute

SVG ANIMATION EXAMPLES

```
<g id="AnimationPaths">
  <path id="Zuerich_Geneva" stroke="orange" stroke-width="2000" fill="none" d="M6825
  <use id="AirplaneZurichGeneva" xlink:href="#airplane">
    <animateMotion id="animMotionZurGen" dur="5s"
      repeatCount="indefinite" rotate="auto-reverse">
      <mpath xlink:href="#Zuerich_Geneva"/>
    </animateMotion>
    <animateTransform attributeName="transform" type="scale"
      keyTimes="0;0.2;0.8;1" values="1.5;4;4;1.5"
      dur="5s" additive="replace" fill="freeze"
      repeatCount="indefinite"/>
  </use>
</g>
```



HTML / CSS TIMING

- No timing model, no concept of
 - Timeline
 - Time container
 - Timed Element
- All animations are independent
 - Triggered either via CSS or JS
 - Loose synchronization

CSS ANIMATIONS

- Declare animations in CSS

- Animation is started (offset) when the style is attached
- Different model from SVG

- Properties animation-*:

name, duration, timing-function, iteration-count, direction, play-state, delay

- Keyframe values

```
div {  
  animation-name: diagonal-slide;  
  animation-duration: 5s;  
  animation-iteration-count: 10;  
}  
@keyframes diagonal-slide {  
  from { left: 0; top: 0; }  
  to { left: 100px; top: 100px; }  
}
```

- <http://dev.w3.org/csswg/css3-animations/>

CSS TRANSITIONS

- CSS declaration of smooth value changes between 2 states
 - <http://dev.w3.org/csswg/css3-transitions/>
- Properties transition-*:
 - property, duration, timing-function, delay
 - Ex: values before and after the animations are not known to CSS

```
div {  
  transition-property: opacity;  
  transition-duration: 2s;  
}
```

SCHEDULING ANIMATIONS

- Time-based driven animations
 - The animation starts/stops when deadlines expressed using time values are reached
 - Ex: Start the animation at time 10 in the scene time line
 - Ex: Stop the animation at time 25 in the scene time line
- Event-driven animations
 - The animations starts/stops upon reception of events
 - Ex: start the animation when the user clicks on this button
 - Ex: stop the animation when this other animation starts

SCRIPTED ANIMATIONS

- Use of timers and callback functions
 - Ex: using the window object
 - Ex: using an SVGTimer object
 - Ex: using requestAnimationFrame
- Management of the synchronization by the script

ANIMATIONS WITH JS

```
<rect id='R' width="120" height="50" fill="blue">
<script>
function doAnimation(){
  var rect=document.getElementById('R');
  x=x+xincr;
  rect.setAttribute('x', x);
  window.setTimeout("doAnimation()", 10);
}
</script>
```

```
function animloop() {
  requestAnimationFrame(animloop);
  render();
}
```

SYNCHRONIZATION IN ANIMATIONS AND MEDIA

- Time lines
 - Each media element has a clock and an associated time line
 - The document itself has a clock and associated time line
- Synchronization properties describes how all clocks relate
- Clocks can be
 - Independent: No synchronization.
 - Locked: if one is paused, the other is too. If the pace changes (e.g. speed), the other's pace is changed too.
 - Hard synchronization.
 - Tolerance in the synchronization

WEB ANIMATIONS

■ Goals

- Provide a common foundation for CSS and SVG animation
- Provide a high quality script API for animation and synchronization
 - Easily create sophisticated frame-rate independent animations
 - Amenable to hardware acceleration
- Provide a single target for future animation work
- Advance the set of animation features available on the Web platform

■ Single Model with different syntaxes

- JavaScript API
- CSS Syntax
- SVG Syntax

■ <https://dvcs.w3.org/hg/FXTF/raw-file/tip/web-anim/index.html>