SVG Open / Carto.net Developers Conference

Zurich, Switzerland - July 15-17, 2002

Workshop : SVG and Animation (SMIL, script based)

Organizers: **Stefan Goessner, Michel Hirtzler**

FAQ - FAQ - FAQ - FAQ - FAQ - FAQ - FAQ - FAQ - FAQ - FAQ - FAQ - FAQ - FAQ

What are animation elements ?

You can use the animation elements which are defined in the SMIL Animation specification:
'animate' 'set' 'animateColor' 'animateMotion'
SVG specifications add 'animateTransform' and 'mpath' 'keypoints' 'rotate' to animateMotion
SVG also add some attributs to 'path'.

Which object animation element apply ?

Animation element apply to element referenced by xlink:href="....." or to parent node.
This codes give same effect, rectangle move from –300 along x-axis.

```
<rect x="300" y="100" width="300" height="100" fill="rgb(255,255,0)">
<animate attributeName="x" attributeType="XML" begin="0s" dur="9s" from="300" to="0" />
</rect>

<rect id="RectElement" x="300" y="100" width="300" height="100" fill="rgb(255,255,0)"/>
<animate xlink:href="#RectElement" attributeName="x" attributeType="XML" begin="0s" dur="9s" from="300" to="0" />
```

What can be animated ?

For SVG elements, most attributs are animatable.
W3C specifications give animation property for basic data types

Data type	Additive?	animate	set	animate Color	animate Transform	Notes
<angle>	yes	yes	yes	no	no	
<color>	yes	yes	yes	yes	no	Only RGB color values are additive.
<coordinate>	yes	yes	yes	no	no	
<frequency>	no	no	no	no	no	
<integer>	yes	yes	yes	no	no	
<length>	yes	yes	yes	no	no	
<list of xxx>	no	yes	yes	no	no	
<number>	yes	yes	yes	no	no	
<paint>	yes	yes	yes	yes	no	Only RGB color values are additive.
<percentage>	yes	yes	yes	no	no	
<time>	no	no	no	no	no	
<transform-list>	yes	no	no	no	yes	Additive means that a transformation is post-multiplied to the base set of transformations.
<uri>	no	yes	yes	no	no	
All other data	no	yes	yes	no	no	

Which objects can be animated with animateMotion ?

The following lists all of the elements which can be animated by an animateMotion element: 'g' 'defs' 'use' 'image' 'switch' 'path' 'rect' 'circle' 'ellipse' 'line' 'polyline' 'polygon' 'text' 'clipPath' 'mask' 'a' 'foreignObject'

What are parameters for all animation elements ?

restart : "always" | "whenNotActive" | "never" "always" is the default value
repeatCount : numeric value | "indefinite" value must be greater than 0
fill : "freeze" | "remove" "remove" is the default value
calcMode : "discrete" | "linear" | "paced" | "spline" "linear" is the default value except for "animateMotion" (“paced" is the default value)
dur: Clock-value | "media" | "indefinite" for Clock-value unit by default is second
attributeType : "CSS" | "XML" | "auto" "auto" is the default value
attributeName : name of the target attribute

When start animation ?

begin : begin-value-list
begin-value : (offset-value | syncbase-value | event-value | repeat-value | accessKey-value | wallclock-sync-value | "indefinite")
offset-value : the element begin as an offset from an implicit syncbase, in SVG syncbase begin with document. (in seconds)
syncbase-value : (Id-value "." ("begin" | "end")) animation begin is defined relative to the begin or active end of another animation
event-value : (Id-value ".")? (event-ref) animation begin is defined relative to the time that the event is raised
repeat-value : (Id-value ".")? "repeat(" integer ")" Describes a qualified repeat event. The element begin is defined relative to the time that the repeat event is raised with the specified iteration value.
accessKey-value : "accessKey(" character ")" Describes an accessKey that determines the element begin.
wallclock-sync-value : wallclock(" wallclock-value ")" Describes the element begin as a real-world clock time.
indefinite : begin of the animation will be determined by a "beginElement()" method call or a hyperlink targeted to the element
Example : begin="0;15;id_other_animation.end;accessKey(a);id_button.click"

How start animation by click on button ?

Three examples:

1) begin="go.click"
```
<animate begin="go.click" ..... />
<rect id="go" x="310" y="430" width="50" height="18" opacity="0.3"/>
```

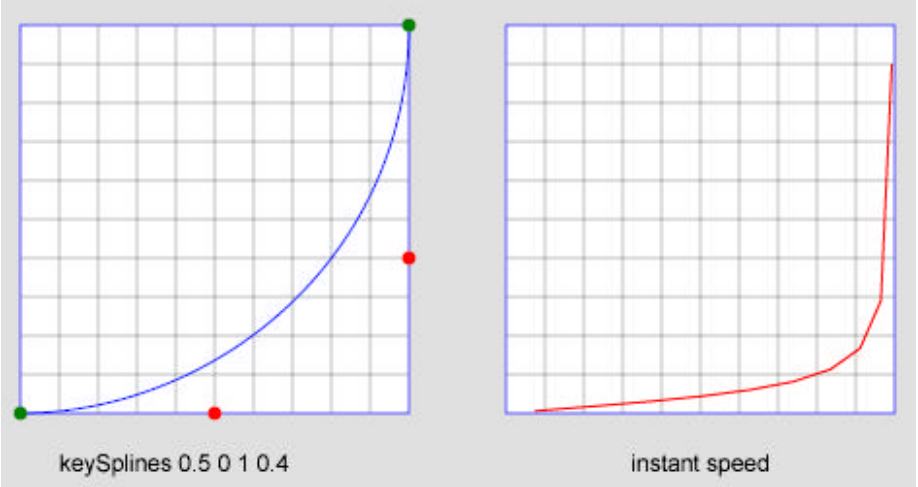
2) begin="indefinite" and hyperlink on button
```
<animate id="a1" begin="indefinite" ..... />
<a xlink:href="#a1">
<rect x="310" y="430" width="50" height="18" opacity="0.3"/>
</a>
```

3) begin="indefinite" and beginElement() method
animate element must be javascript variable, put this code in init function on loading SVG:

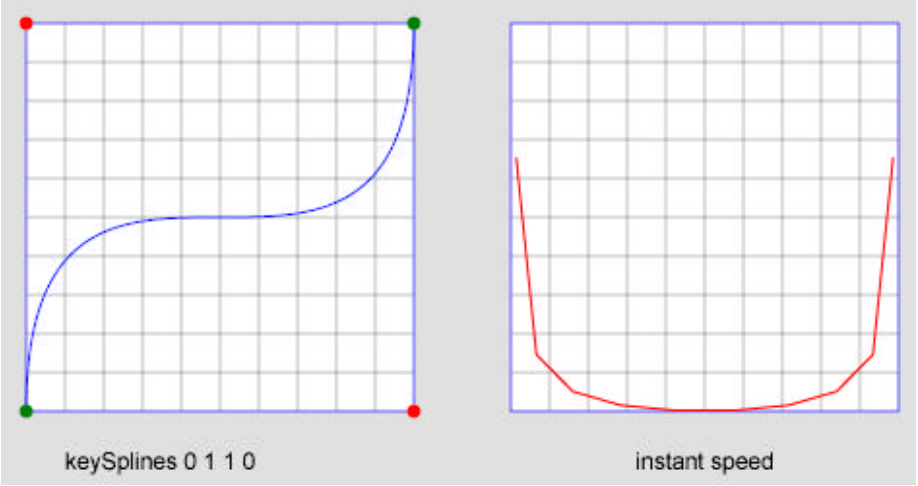
```
svgdoc=evt.getTarget().getOwnerDocument();
a1=svgdoc.getElementById('a1');

<animate id="a1" begin="indefinite" ..... />
<rect onclick="javascript:a1.beginElement()" x="310" y="430" width="50" height="18" opacity="0.3"/>
```

How use keySplines ?

With keySplines you can control speed of animation



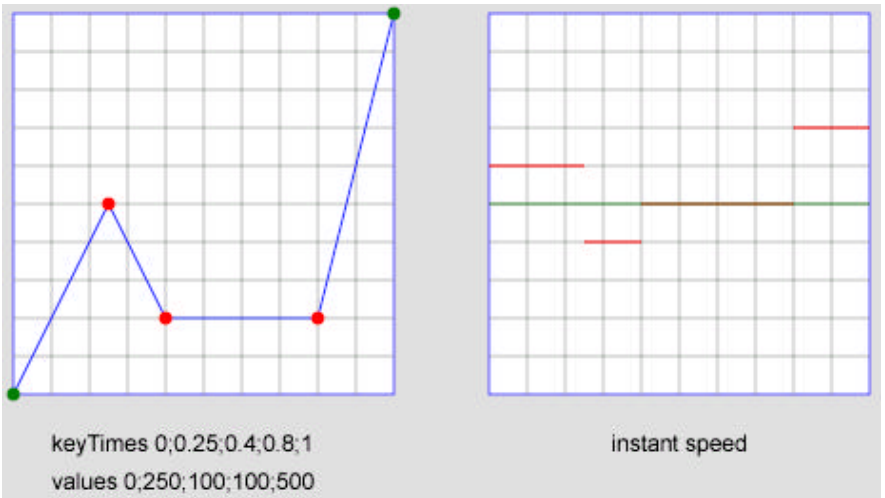
With this example, keySplines=”0.5 0 1 0.4” animation begin slowly and speed increase during animation



In this case, keySplines=”0 1 1 0”, speed decrease, animation stop and speed increase

How use keyTimes ?

With ‘keyTimes’ associated to ‘values’ you can divide animation in sequences.



In this example, with keyTimes=”0;0.25;0.4;0.8;1” and values=”0;250;100;100;500”
if animation apply to width of a rectangle, we will see:
width increase from 0 to 250 during 25% of duration, then width decrease to 100 during 15% of duration, width stay to 100 during 40% and increase to 500 during last 20%.

For each sequence, you can define speed with ‘keySplines’, in this example you can add keySplines=”0.5 0 1 0.4;0 0.5 0.5 1;0 1 1 0;0.5 0 1 0.4” to have speed increase on first sequence, decrease on second and increase on last.

How start animation with keyboard ?

With begin="accessKey(a)", animation start when user tape 'a' on keyboard.
Caution: SVG must have focus, you can move mouse or click on SVG to get focus.

How produce progressive drawing ?

We have to use stroke-dasharray and stroke-dashoffset and know length of curve to draw.
This code draw nothing on screen for circle, ellipse, polyline, polygone or path :

```
<... style="stroke-dasharray:curve_length, curve_length;stroke-dashoffset:curve_length"/>
```

If in animate element stroke-dashoffset decrease to 0, we get progressive drawing of curve.

```
<circle cx="200" cy="200" r="115" style="fill:none;stroke:blue;stroke-dasharray:723, 723;stroke-dashoffset:723">
<animate begin="0" attributeName="stroke-dashoffset" from="723" to="0" dur="5s" fill="freeze"/>
</circle>
```

How calculate length of curve ?
For a path, getTotalLength() give length of curve, for others we have to calculate length in script.
Loading SVG, function init put value in curve and animation

```
function init(evt,curve_id,animation_id)
{svgdoc=evt.getTarget().getOwnerDocument();
curve=svgdoc.getElementById(curve_id);              // get curve object
length_curve=curve.getTotalLength();                // get length of curve
// curve is not visible for this values in stroke-dasharray and stroke-dashoffset
curve.getStyle().setProperty('stroke-dasharray', length_curve.toString()+","+length_curve.toString());
curve.getStyle().setProperty('stroke-dashoffset', length_curve.toString());
// put value in animate element
animation=svgdoc.getElementById(animation_id);
animation.setAttribute('from', length_curve.toString());
}
```

When obtain morphing with "d" attribute ?

If we change "d" in a path, we have morphing along animation in some cases (in ASV3):
Bezier's cubic or quadratic, moveto and lineto with same number of points in all values and no arcs.

How create fade-in out between pictures ?

Using animation on feComponentTransfer and alpha channel, when slope go from 1 to 0, picture is cleared, and with animated filter on each picture, we get fade-in out.
We can add effects with feDisplacementMap and radial gradient as grid by example, animating scale attribute.

How animate letters in a text ?

If we want to see each letter of text animated, we can use javascript to create SMIL animation.
Each letter has to be centered for using transform attribute and put in place with translation.
We add animation element for each letter.
To create 3D effect, letter rotate around axis, we use javascript to create animation, modifying matrix of transformation.

This function put each letter of text in place, we can enter any text in text_string:

```
function init(evt,text_string)
{svgdoc=evt.getTarget().getOwnerDocument();

// create text element to get position of each letter

node=svgdoc.createElement("text");
node.setAttribute("x","100");node.setAttribute("y","100");
node.setAttribute("style","visibility:hidden;text-anchor:middle;font-size:80;font-family:Arial;fill:black");
node.setAttribute("id","text");
texte=svgdoc.createTextNode(text_string);
node.appendChild(texte);
where=svgdoc.getElementById('text_place');
where.appendChild(node);

// get position and create each glyphe

text_object =svgdoc.getElementById('text');
for (i=0;i<text_string.length;i++)
{f= text_object.getExtentOfChar(i);
node=svgdoc.createElement('g');
node.setAttribute("transform","translate("+(f.x+f.width/2)+","+(f.y+f.height)+")");
node2=svgdoc.createElement('text');
node2.setAttribute("id","letter"+i.toString());

// letter is centered for rotate around vertical axis

node2.setAttribute('x',-f.width/2);
node2.setAttribute('y',"0");
node2.setAttribute('style','text-anchor:left;font-weight:bold;font-size:80;font-family:Verdana;fill:red;stroke:black');

// add matrix for 100% javascript animation or add animation element as child

node2.setAttribute('transform','matrix(1 0 0 1 0 0)');
text_letter=svgdoc.createTextNode(chaine.substring(i,i+1));
node2.appendChild(text_letter);
node.appendChild(node2);
where.appendChild(node)
}};
```

How move spot on picture or text ?

We can use clipPath and animateMotion on a path for shape that define clipPath.

How create streaming of pictures ?

We can use cliPath, animate coordinates of area and to keep picture in same place add animateTransform using clip-path.

What is difference between 'animate' and 'set' element ?

'Set' element give no progressive effect, it don't use attributes as calcMode, keyTimes, keySplines ...
By example with this animation apply to rectangle with 50 as width:
<set attributeName="width" to="200" dur="3s"/>
Width of rectangle is 200 to 0s, and at 3s 50.
To keep width of 200, you can add fill="freeze"

How use javascript to create animation ?

First method: recursive call to function with setTimeout (function , duration)

```
// define function animate as window method
window.animate=animate

function animate(evt) {
// quit function animate() on condition

if (condition) { .....  return };
// action: modify SVG objects
.....
// recursive call to animate() after duration in milliseconds
setTimeout ( "animate()" , duration )
};
```

Second method: call to function with setInterval (function , duration)

```
function animate(evt) {
// call to function modifying SVG objects each duration in milliseconds
setInterval ( "rebuild()" , duration )
}
function rebuild(evt) {
// action: modify SVG objects
.....
}
```

How start or close animation with javascript ?

begin="indefinite" and beginElement() or endElement() method
animate element must be javascript variable, put this code in init function on loading SVG:

```
svgdoc=evt.getTarget().getOwnerDocument();
a1=svgdoc.getElementById('a1');

<animate id="a1" begin="indefinite" ..... />
To start animation by click on object:
<rect onclick="javascript:a1.beginElement()" x="310" y="430" width="50" height="18" opacity="0.3"/>
To close animation
<rect onclick="javascript:a1.endElement()" x="310" y="430" width="50" height="18" opacity="0.3"/>
```