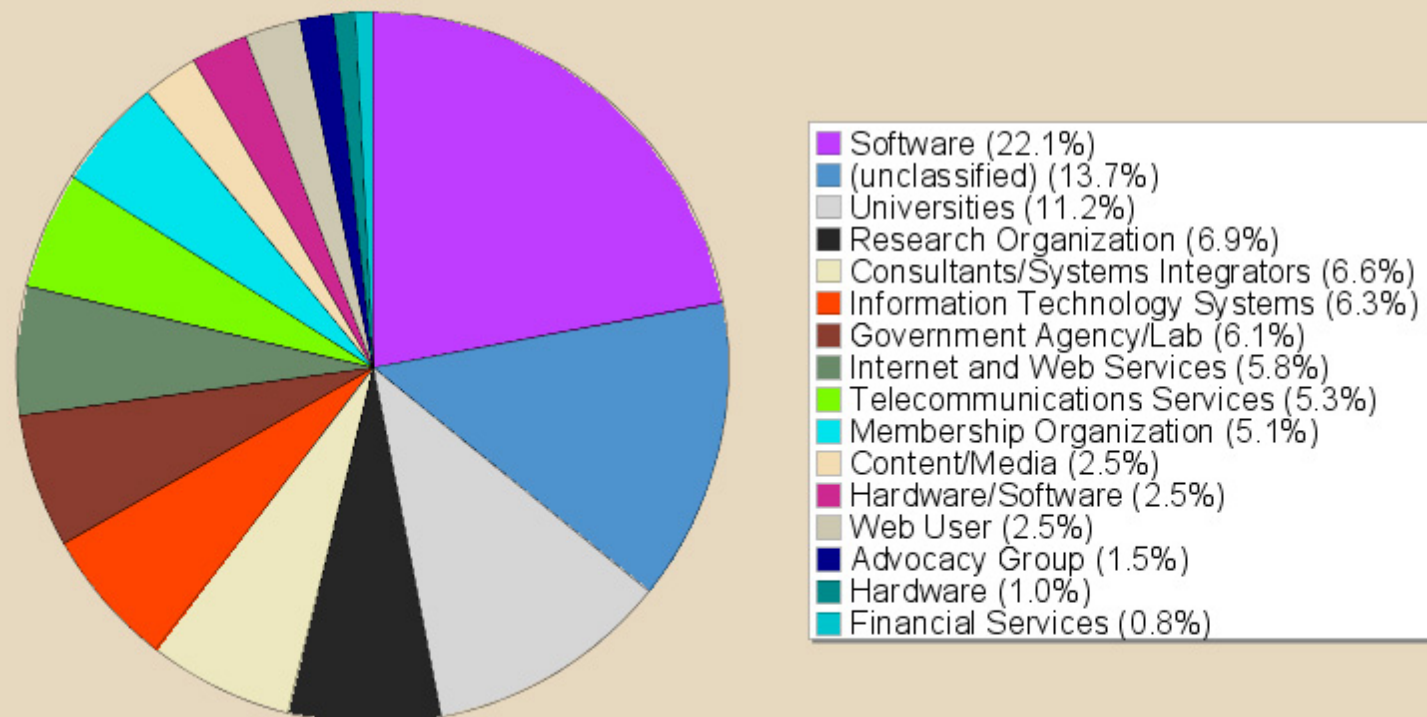


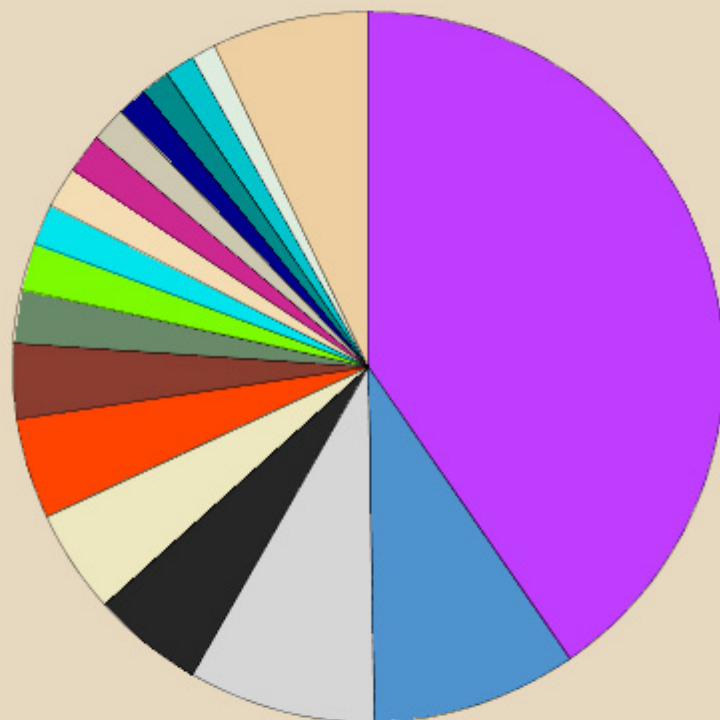
“To lead the World Wide Web to its full potential by developing protocols and guidelines that ensure long-term growth for the Web”

- Founded by Tim Berners-Lee in 1994
- Develops open *Recommendations* (Web Standards)
 - over eighty so far...
- Also engages in education, outreach, develops guidelines...
- *A neutral forum for building consensus around Web standards*

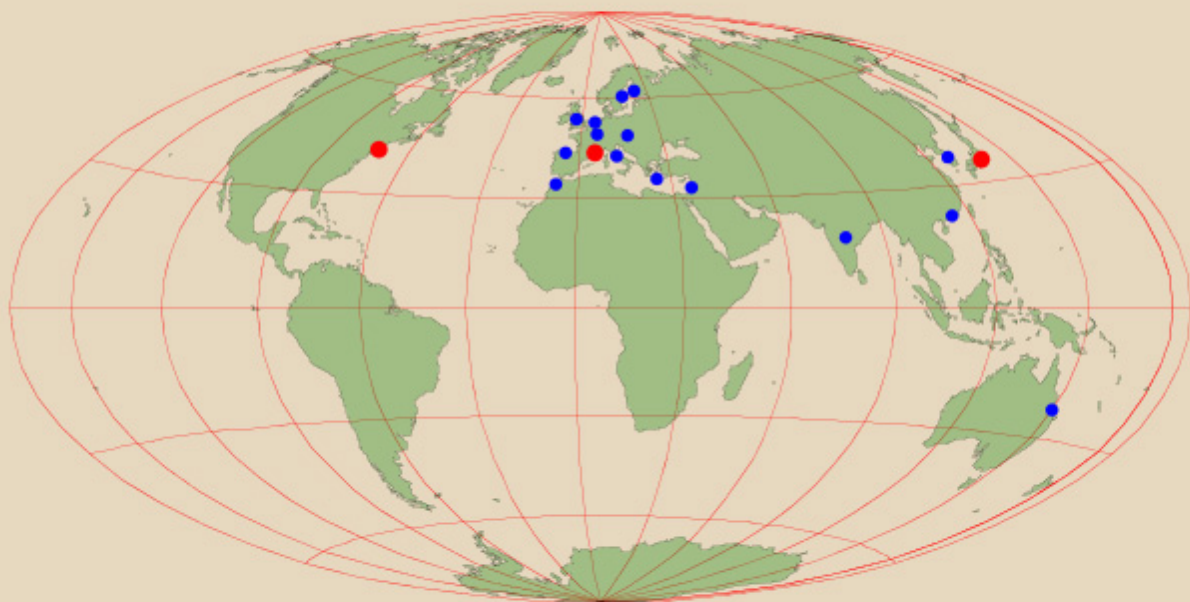
- Web Technologies should be interoperable
 - the Web is based on a large palette of technologies
 - no technology can pretend to cover *all* needs on the Web
 - hence the *interoperability* of technologies necessary
- Web Standards should be open, i.e., non-proprietary
- The Web should be accessible to *all*

- W3C has around 400 members
 - companies, research institutes, universities, government agencies
 - work is done by working groups staffed by members





United States	(40.4%)
Japan	(9.3%)
United Kingdom	(8.5%)
Germany	(5.1%)
Spain	(4.8%)
Italy	(4.5%)
France	(3.5%)
Canada	(2.4%)
Netherlands	(2.1%)
Finland	(1.9%)
Sweden	(1.9%)
Korea	(1.9%)
Hong Kong SAR, China	(1.6%)
Ireland	(1.3%)
Israel	(1.3%)
Australia	(1.3%)
Greece	(1.1%)
Other	(7.2%)



Applications



XML, Namespace, Schema, X Query/XPath/XSLT, DOM, XML Base, X Pointer, RDF/XML, SPARQL

XML Infoset, RDF Graph

Web Architectural Principles

URI/IRI/HTTP

The Web Advancing to its Full Potential

Internet

- Pixel based images (gif, png, jpeg, ...)
 - Fixed resolution
 - Can be very large
 - Original “information” is lost
 - difficult to add metadata
 - difficult to adapt to viewing environment
 - Limited interaction (except for animated gifs and the like)
 - Client/server side image maps for linking
 - clumsy, requires separate authoring tools
- Other, proprietary solutions
 - proprietary authoring tools, players, ...

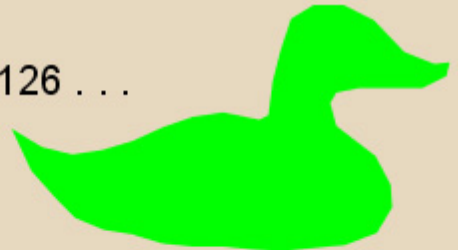
- Vector graphics (i.e., describe geometry, not pixels)
 - resolution independent
 - keeps “content” information
 - but should allow the “inclusion” of, e.g., photographs
- Textual encoding (too)
 - searchable
 - easy to edit, add links, etc
 - makes “programming by example” easy...
- Based on XML
 - to integrate with the rest of the Web

Scalable Vector Graphics (SVG):

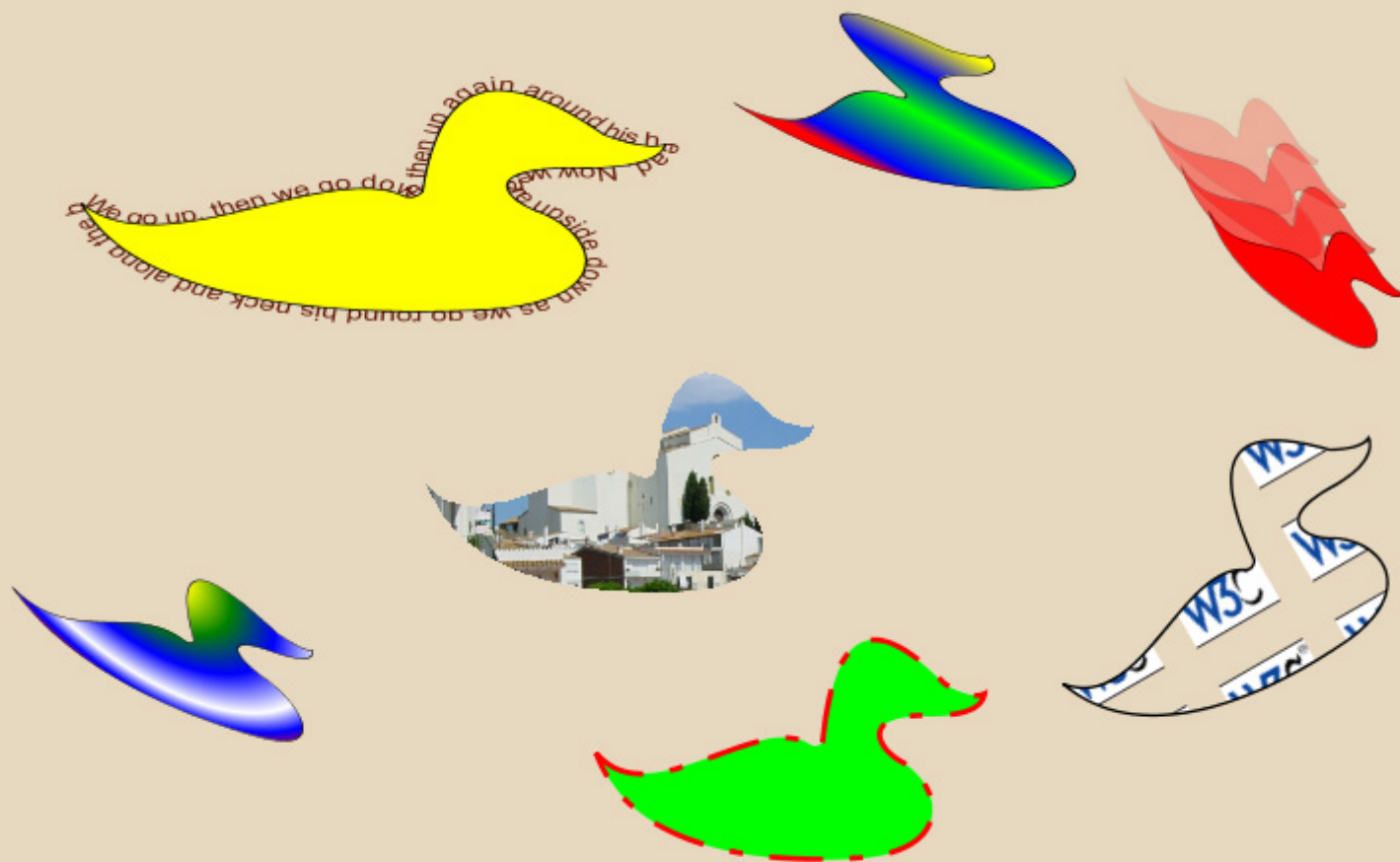
- W3C Working Group was created in 1998
- Specification quite stable by 2000, with first implementations
- Recommendations:
 - *SVG 1.0 (September 2001)*
 - *SVG 1.1 and SVG Mobile (January 2003)*
- SVG 1.1 and SVG 1.0 are functionally identical
 - SVG 1.1 is a “modularization” of SVG 1.0
- Work is continuing towards SVG 1.2 (and SVG 2.0 ...)

“SVG is the HTML for Graphics”

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN"
"http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
<svg width="600" height="300">
  <g transform="translate(10 10)">
    <g stroke="none" fill="lime">
      <path d="M 0.0 112 L 20 124 L 40 129 L 60 126 ...
      ...
    </path>
  </g>
</svg>
```



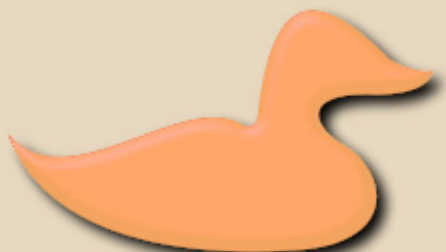
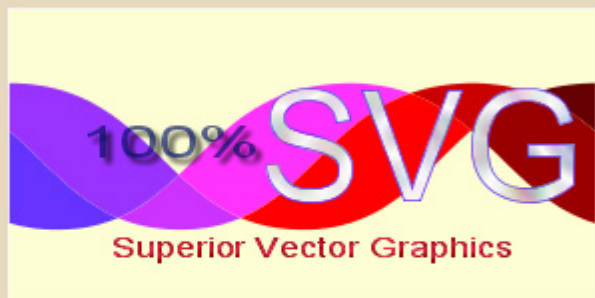
- Paths can:
 - be closed or open
 - define arcs, quadratic and cubic splines, ...
 - be filled with colour, gradients, patterns, ...
 - be shortcut for rectangles, circles, ...
- Attributes can control line style, dash, joining, miter, ...
- Bitmap images can be included
- Clipping, masking
- Separate text element with tons of attributes
 - e.g., to ensure internationalization (right to left or vertical writings)
- Graphics can be nested and/or referred to as “symbols”
- Everything can be transformed
 - rotated, skewed, translated, etc
- *All the “usual” operations of 2D graphics are available*

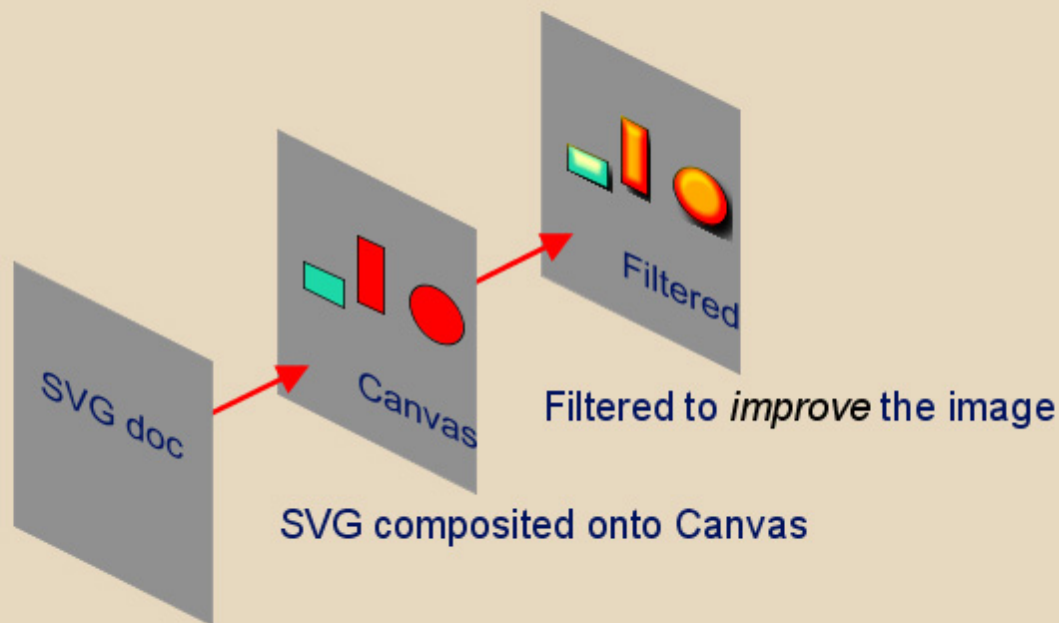


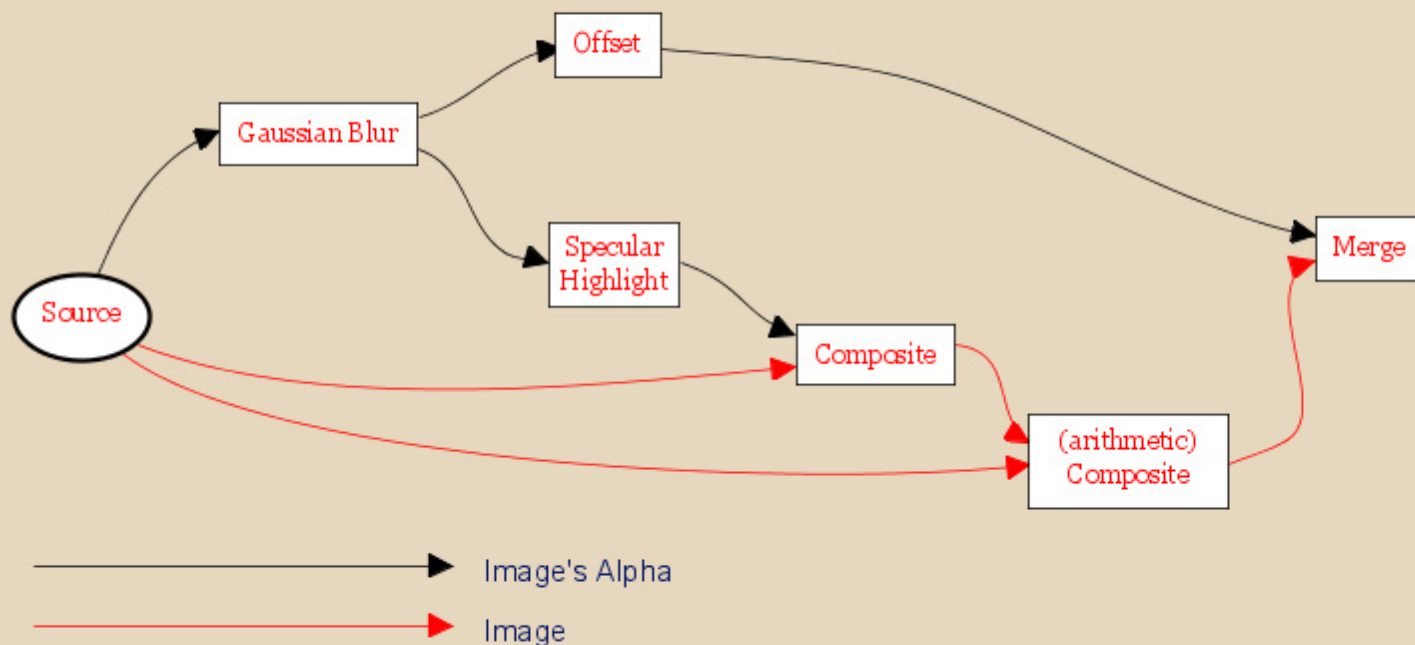
- Q: Is SVG any more than “PostScript in XML”?
(or: is SVG interesting for a specialist in Computer Graphics?)
- ▶ A: very much so!
 - SVG includes a number of “extra” features:
 - filter operations
 - animations
 - scripting
 - CSS type styling
 - SVG Fonts
 - it is also a good fast prototyping tool for 2D graphics
 - can be (and is!) used for graphics courses, for example

- *SVG is integrated*
 - SVG is not a “technology island”, but...
 - ... it is an XML applications that is well integrated with the Web
- What this means is that:
 - SVG relies on a series of other specifications
 - XML tools (editors, parsers, validators) can be used directly
 - some of the general W3C concerns are valid to SVG, too

- Graphics on the Web is often used for “artwork”
 - complex logos
 - “sexy” presentations (shadow effects, pseudo 3D)
 - advertisements
- *Filters* have been added to SVG to produce such artworks:







- ▶ Colour manipulation
 - direct colour matrix manipulation
 - component transfer (brightness, contrast/gamma control, ...)
- ▶ Compositing
 - merging, blending
 - general compositing
- ▶ Picture modification
 - convolution (blurring, sharpening, etc)
 - fattening/thinning (i.e., dilation or erosion)
 - (Gaussian) blur
 - offsetting
 - displacement maps
 - displace pixels in one image under the control of another
- ▶ Picture creation
 - generation of artificial textures (turbulence functions)

Full 3D Effect (like before):



Composite pixel function: "over", "xor", etc,
or "arithmetic" with result = $k1*i1*i2 + k2*i1 + k3*i2 + k4$

```
<filter id="composite" ...>
```

```
  <felmage xlink:href="im1.jpg" result="i1">
```

```
  <felmage xlink:href="im2.jpg" result="i2">
```

```
  <feComposite in="i1" in="i2" operator="arithmetic" k1=".0" k2="1" k3=".0" k4=".0"/>
```

```
</filter>
```

...

```
<rect fill="none" filter="url(#composite)"/>
```



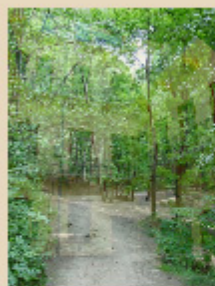
k2="1" k3="0"



k2=".75" k3=".25"



k2=".5" k3=".5"



k2=".25" k3=".75"



k2="0" k3="1"

Convolution matrix over pixels

```
<filter id="blur">
```

```
  <feConvolveMatrix order="3" kernelMatrix="3 0 3 0 0 0 3 0 3"/>
```

```
</filter>
```

```
...
```

```
<image ... filter="url(#blur)"/>
```



blurring matrix =

$$\begin{bmatrix} 3 & 0 & 3 \\ 0 & 0 & 0 \\ 3 & 0 & 3 \end{bmatrix}$$


sharpening matrix =

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

- Filters have a strong “image processing” flavour
- Very few graphics system combine these two
 - historically, there hasn't been enough contacts between these two communities...

- SVG Animation = change object attributes dynamically
- Animation is controlled through a set of animation objects
 - reusing parts of W3C's SMIL 2.0 Specification
(Synchronized Multimedia Integration Language Version 2.0)
- *Declarative syntax*, no real increase in file size
- Animation is performed on client side

Interaction
domain

Technology and Society
domain

W3C WORLD WIDE WEB
consortium
Web Accessibility
initiative

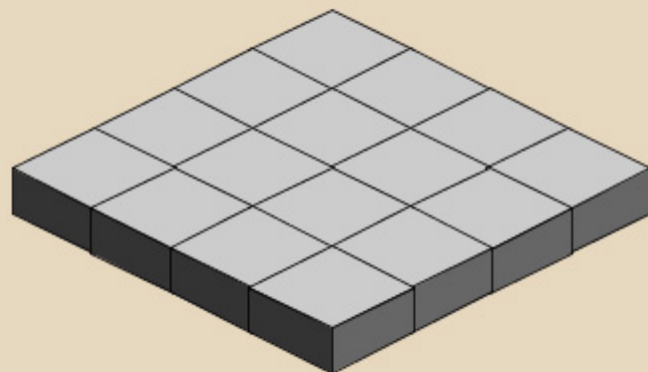
Architecture
domain

- Practically all attributes can be changed
- Separate animation elements
 - `animate`, `animateMotion`, `animateTransform`
 - `animateColor`
 - `set` (changes discrete values)
- Example:

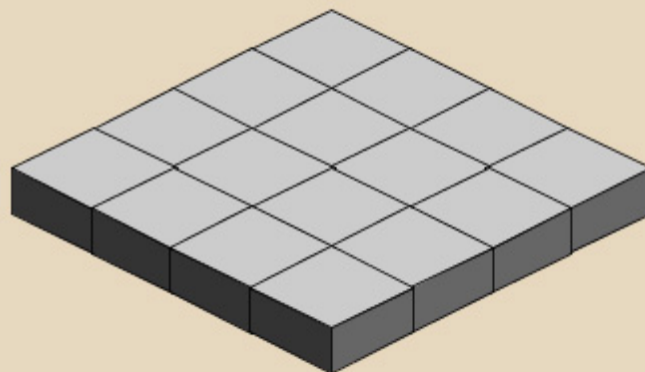
```
<circle id="c1" r="50" ...>  
  <animate attributeName="r" from="50" to="100" ...>  
</circle>
```



- Control the pace of change in values
- Control of the timing of animation steps
 - reacting to events, chained...
- Control on *what* is being animated
- Control of timing
- etc.

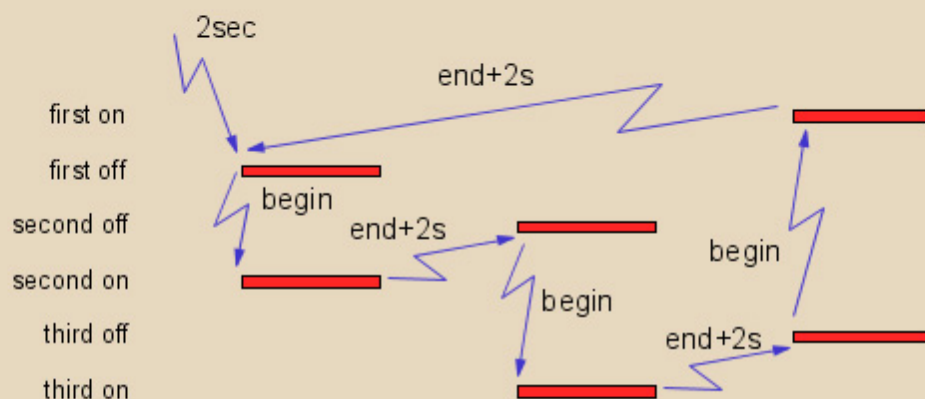


`calcMode="linear"`



`calcMode="spline"`

Copyright © Antoine Quint



Put a light source “above” the image and ...
and animate its attributes!

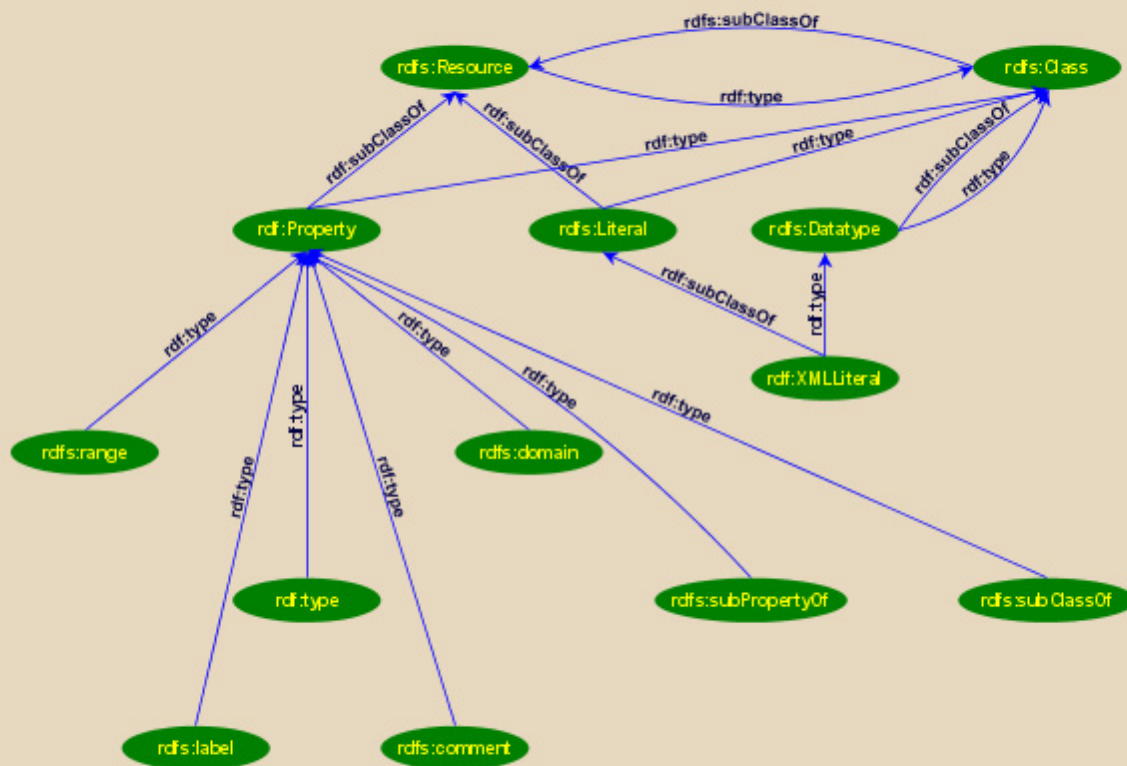


- DOM = Document Object Model
 - programmatic interface to the runtime XML tree
 - *a general W3C mechanism* with an SVG specialization
- Scripts can be included in SVG (through a `script` node)
 - usually ECMAScript (a.k.a. Javascript)
 - sometimes Java
- Scripts can modify:
 - the XML tree (add and/or delete nodes)
 - the attributes of individual objects
- SVG is primarily declarative, but...
- ... this allows for programming *when necessary*

- Numerical calculations
- Access to some system resources (e.g., clock)
- Complex animation and/or interaction
 - complicated visual effects
 - need for the state of interaction
- Random features (e.g., random choice among alternatives)
- Client/server type applications
- Managing application data embedded in the SVG content

```
<g onload="Start(evt)" ... >  
...  
function Start(evt)  
    var now=new Date();  
    clockObj.adjustTime(now);  
}
```

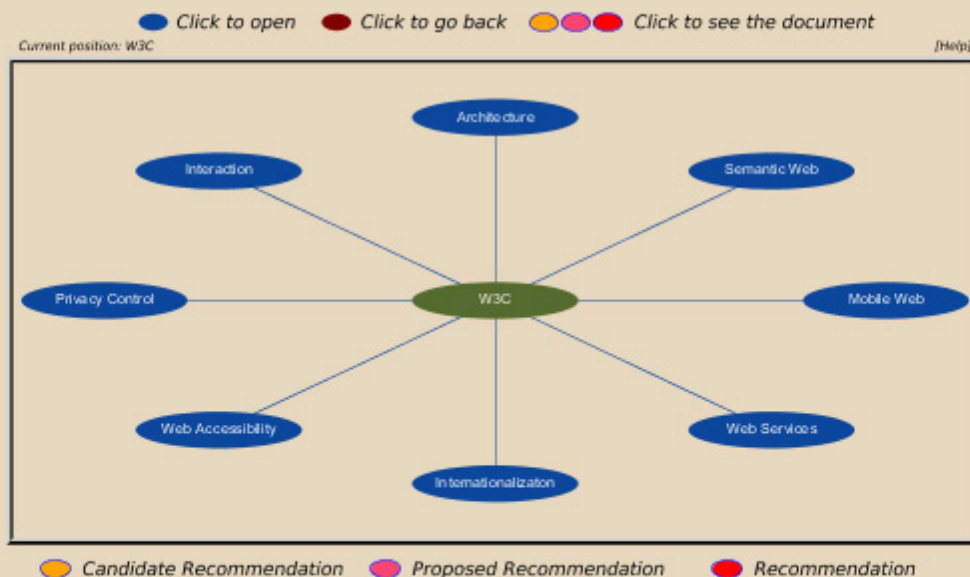




```
<svg ... xmlns:appl="http://..." onload="init(evt)">
  <style xlink:href="preds.js" ... />
  ....
  <appl:resource id="R1" x=".." y=".." label=".."/>
  <appl:resource id="R2" x=".." y=".." label=".."/>
  <appl:predicate subject="R1" object="R2" label=".."/>
  ....
</svg>
```

- A script displays the nodes, calculates the arcs, places the labels on the arcs, etc
 - The DOM includes *all* XML data, not only SVG
 - the SVG player ignores the non-SVG ones visually...
 - ...but those are visible to a script!
- *SVG turns into an application development environment!*

- Similar mechanism (user data mixed with SVG)
 - if you know how to (ECMA)script an HTML page...
... you (almost 😊) know how to script SVG!



Japanese:
W
e
b
の
可
能
性
を
最
大
限
に
導
き
出
す
た
め
に
:

English: Leading the Web to its Full Potential...
Catalan: Duent la Web al seu ple potencial...
Dutch: Het Web tot zijn volle potentieel ontwikkelen...
French: Amener le Web vers son plein potentiel...
German: Alle Möglichkeiten des Web erschließen...
Greek: Οδηγώντας τον παγκόσμιο ιστό στο μέγιστο των δυνατοτήτων του...
Hungarian: Hogy kihasználhassuk a Web nyújtotta összes lehetőséget...
Italian: Sviluppare al massimo il potenziale del Web...
Chinese: 引发网络的全部潜能...
Korean: 웹의 모든 잠재력을 이끌어 내기 위하여...
Portuguese: Levando a Web em direção ao seu potencial máximo...
Russian: Раскрывая весь потенциал Сети...
Spanish: Guiando la web hacia su máximo potencial...
Swedish: Se till att Webben når sin fulla potential...
Finnish: Ohjaamassa Webin kehittymistä täyteen mittaansa...
Hebrew: להוביל את הרשת למיצוי הפוטנציאל שלה...
Arabic: لإيصال الشبكة المعلوماتية إلى أقصى إمكاناتها...

Chinese:
引
發
網
絡
的
全
部
潛
能
:

- Extremely dynamic market
- Big business in Europe and Asia, with US catching up fast
- Potentially huge number of users
 - 40 Million new users per year in China alone!
 - future: one PC per family, but one Mobile *per person*...



- SVG has “smaller” versions: *Basic* and *Tiny*
- SVG Tiny is the official vector graphics format for 3GPP
- Has become *the* vector graphics tool for Mobiles
 - on Windows CE, Symbian, for J2ME...
 - a number of phones come with SVG built-in
 - integration browsers is also happening



- Exciting additions:
 - inclusion of video and audio
 - print and streaming control
 - complex compositions
 - “take the ‘xor’ of the graphics with the content of the canvas”
 - ...
- Multiresolution content
- Lots of small improvements
- Some features already implemented in latest Adobe release

These slides:

<http://www.w3.org/2005/Talks/0830-Leuven-IH/>

Mail me if you want:

ivan@w3.org