Source: ENST and France Telecom
Status: Proposal
Title: Support of SVG1.2 for LASER
Authors: Jean-Claude Dufourd, Olivier Avaro and Cyril Concolato

Introduction
It is important to leverage industry’s growing acceptance of SVG to improve LASER chances of success. In particular, in the mobile arena, 3GPP mandates the use of SVG Tiny in its releases 5 and 6 within the framework of MMS. Thus, SVG is seen by many actors as a must, and supporting a subset of SVG in LASER would be a great asset.

Yet SVG has the same problems as BIFS in terms of size and performance. Even though one SVG profile is called Tiny, its implementations are well above the target range of LASER:

- BitFlash website: SVG Tiny player on mobiles usually around 500K (870K with scripting).
- CSIRO player for PocketPC (from W3C website): 390K
- eSVG player for PocketPC: around 500K (plus maybe DLLs)

With hindsight, BIFS was not so bad with IBM’s Java player around 400K (including video and audio decoding).

There is at [www.tinyline.com](http://www.tinyline.com), a Java implementation of a small subset of SVG, called Tinyline (version 1.6). The author calls the implemented subset SVG Minute. 30 of the 32 elements of SVG Tiny are implemented, and definitely not all attribute combinations. The result is still of amazingly good quality, with a code size of 168K (class files = memory usage in the handheld), and 95.1K (zipped in a JAR file = size of installer message when sent over the air). Our evaluation of the code is that its quality is very high, and we believe it is an excellent reference point.

To achieve the LASER target of 50-100K, there is a definite need for drastic profiling of SVG Tiny. By comparison and to prove this target is feasible, our proto-LASER Java player has a code size of 79.5K(classes) and 42.7K(zipped).
SVG Complexity and Profiling

Estimation of the Problem Scale

As there was “node bloat” in BIFS, there seems to be “feature bloat” in SVG, already in the so-called “Tiny” profile. In this document, we propose a way of measuring the complexity of the standard. In SVG Tiny, there are 32 XML elements (nodes in BIFS), with an average 20 XML attributes (fields in BIFS) per element. The total size is close to 700 allowed element-attribute pairs in total. The list is given in an annex to this document. No wonder the code size in the 500K range.

By comparison, our proto-LASER solution has 30 element-attribute pairs for SAF and 180 for LASER, including 30 for BIFS-like updates and 150 for scene.

So the target is a feature reduction of 4 to 8.

Binary vs. XML Text

Our proto-LASER implementation includes a binary parser à la BIFS: clearly, binary parsing is smaller and faster than XML parsing. Here is an estimation of the difference in size.

In Tinyline, the parser size is approximately 66K (classes) and 30K (zipped). This accounts for an incomplete SVG Tiny (XML) parser, without DOM or scripting.

In our proto-LASER implementation, the parser size is 34.1K (classes) and 14.6K (zipped).

Clearly, with a factor of 2 in code size, and a significant speed up (not measured), a binary format is much better for small embedded devices.

SVG Tiny Features

SVG is loaded with authoring facilities and features useful in text but painful in binary. These features are definitely assets in the PC/Internet world, where easy textual authoring is a key to the acceptance of a format. The mobile world does not have these requirements, and can even be said to have conflicting requirements. Here is a list of problematic features:

- switch: the SVG switch is a construct that enables a varying “level of requirements” within one scene. While this is a very handy capability for device independence, it is too complex for embedded devices.

- property inheritance: the ability to place e.g. a color on the primitives or on any of the elements above it is again useful in terms of authoring, but multiplies the number of possibilities, hence the complexity of the code to implement the “color with inheritance” feature.

- multiple primitives when path/shape can do it all: having “rect” and “path” is not useful on embedded devices, because what “rect” does, “path” can do very easily. Thus, only “path” is needed as a graphics primitive.

- foreignObject: SVG has an element for including “something else” in the scene tree. This is as close as ApplicationWindow node of BIFS as we can think. This could intended to be used for e.g. video. Clearly, MPEG (and SMIL) has another way of doing it, by defining Audio and Video elements.

- complete text rendering system: the complexity of the text rendering systems is big. We have recently acknowledged within MPEG the need for better text, but the SVG text level is still overkill for the smallest devices.
- multiple ways of doing something: SVG does not have the “one-tool-one-functionality” principle that MPEG has, and this can be felt throughout the spec.
- in SVG 1.2: the SVG WG has decided to put simple gradients and text flow layout in SVG Tiny, something that brings SVG Tiny even further than the target code size.
- path specification with all those options: the multiple types of curves that path can do are overkill on small devices.
- multiple transform specifications: the ability to specify transformations in multiple ways has a high cost in lines of codes
- color keywords: the ability to specify colors by keyword is very expensive in terms of 1) code size: the table needed to convert the names to RGB and 2) compression: the encoding of strings is worse than the encoding of RGB.
- defs: this grouping of the DEFed elements is “nice” but does not help code size or implementability in any way, because it is optional. Moreover, it can be done with existing nodes. Therefore, we want to remove it.

**Required Extensions**

To accommodate some of MPEG assets in graphics, here are proposed extensions of SVG:

- Audio and Video elements
- updates à la BIFS: Insert/Replace/Delete, value/node/indexed… to replace the “discarded” XML Events.
- cursor, textual input and keyboard management:
  - cursor: for terminals without pen or mouse (the majority), the ability to designate an actual shape in the content to be a virtual cursor is very effective. This virtual cursor/shape is to be moved by key presses (e.g. with the joystick which emulates key presses), and when the “FIRE” button is pressed, the current position of the virtual cursor is used instead of a mouse position.
  - textual input: this is an interface to the typing assistance software available in most terminals.
  - keyboard management: basically, this is the BIFS KeySensor (specialisation of the InputSensor).
- declarative scripting: as programmatic scripting is too expensive for the target code sizes, we suggest to use BIFS-like scripting, based on concepts close to Conditional, Valuator …
- the ability to save and restore small chunks of data in an application-linked manner has been found key to many mobile services. This should be done in the manner of the RMS feature of MIDP1.0.

**Proposed Profile**

To be discussed with SVG experts, here is a first shot at pruning SVG Tiny. For easier reading, we have used color coding in the annex describing all of the SVG Tiny 1.1 features:

- red/italics means not included
- blue/plain means under consideration for a higher profile of LASER
- black/bold means proposed for inclusion in LASER.

**Acknowledgements**

Some of the work leading to this contribution was funded by the European Commission as part of the DANAE Project.
Annex: all of SVG Tiny allowed element-field pairs

defs container for elements with ids
* .id DEF name
*.xml:base change base for relative urls
*.xml:lang language identifiers
*.xml:space preserve white space
defs .requiredFeatures
defs .requiredExtensions
defs .systemLanguage
defs .transform
desc textual description of the content
* .id DEF name
*.xml:base change base for relative urls
*.xml:lang language identifiers
*.xml:space preserve white space
g grouping node
* .id DEF name
*.xml:base change base for relative urls
*.xml:lang language identifiers
*.xml:space preserve white space
g .requiredFeatures
g .requiredExtensions
g .systemLanguage
g .transform
metadata “MPEG-7” element
* .id DEF name
*.xml:base change base for relative urls
*.xml:lang language identifiers
*.xml:space preserve white space
svg top node
* .id DEF name
*.xml:base change base for relative urls
*.xml:lang language identifiers
*.xml:space preserve white space
svg .requiredFeatures
svg .requiredExtensions
svg .systemLanguage
svg .x
svg .y
svg .width
svg .height
svg .viewBox
svg .preserveAspectRatio
svg .zoomAndPan
svg .version
svg .baseProfile
title title of the content
* .id DEF name
*.xml:base change base for relative urls
*.xml:lang language identifiers
*.xml:space preserve white space
use reference to an ID-ed element
* .id DEF name
The 'switch' element evaluates the requiredFeatures, requiredExtensions and systemLanguage attributes on its direct child elements in order, and then processes and renders the first child for which these attributes evaluate to true. All others will be bypassed and therefore not rendered.
line.stroke-width
line.color-rendering
line.display
line.visibility
line.x1
line.y1
line.x2
line.y2
line.transform

path
* .id             DEF name
* .xml:base       change base for relative urls
* .xml:lang       language identifiers
* .xml:space      preserve white space
path.requiredFeatures
path.requiredExtensions
path.systemLanguage
path.transform
path.d
path.pathLength
path.color
path.fill
path.fill-rule
path.stroke
path.stroke-dasharray
path.stroke-dashoffset
path.stroke-linecap
path.stroke-linejoin
path.stroke-miterlimit
path.stroke-width
path.color-rendering
path.display
path.visibility

polygon
* .id             DEF name
* .xml:base       change base for relative urls
* .xml:lang       language identifiers
* .xml:space      preserve white space
polygon.requiredFeatures
polygon.requiredExtensions
polygon.systemLanguage
polygon.color
polygon.fill
polygon.fill-rule
polygon.stroke
polygon.stroke-dasharray
polygon.stroke-dashoffset
polygon.stroke-linecap
polygon.stroke-linejoin
polygon.stroke-miterlimit
polygon.stroke-width
polygon.color-rendering
polygon.display
polygon.visibility
polygon.points
polygon.transform

polyline
* .id             DEF name
image.xlink:show
image.xlink:actuate
image.requiredFeatures
image.requiredExtensions
image.systemLanguage
image.preserveAspectRatio
image.color
image.fill
image.fill-rule
image.stroke
image.stroke-dasharray
image.stroke-dashoffset
image.stroke-linecap
image.stroke-linejoin
image.stroke-miterlimit
image.stroke-width
image.color-rendering
image.display
image.visibility
image.overflow
image.transform
image.x
image.y
image.width
image.height

text
*.id DEF name
*.xml:base change base for relative urls
*.xml:lang language identifiers
*.xml:space preserve white space
text.requiredFeatures
text.requiredExtensions
text.systemLanguage
text.transform
text.x
text.y
text.rotate
text.color
text.fill
text.fill-rule
text.stroke
text.stroke-dasharray
text.stroke-dashoffset
text.stroke-linecap
text.stroke-linejoin
text.stroke-miterlimit
text.stroke-width
text.color-rendering
text.font-family
text.font-size
text.font-style
text.font-weight
text.display
text.visibility
text.text-anchor
text.alignment-baseline
text.baseline-shift
text.direction
text.dominant-baseline
text.glyph-orientation-horizontal
glyph                          graphics for a given character
*.id                            DEF name
*.xml:base                       change base for relative uris
*.xml:lang                       language identifiers
*.xml:space                      preserve white space
glyph.unicode
glyph.glyph-name
glyph.d
glyph.arabic-form
glyph.lang
glyph.horiz-adv-x

hkern                           kerning pairs and adjustment values in the hor. advance value
*.id                            DEF name
*.xml:base                       change base for relative uris
*.xml:lang                       language identifiers
*.xml:space                      preserve white space
hkern.g1
hkern.g2
hkern.u1
hkern.u2
hkern.k

missing-glyph                   graphics to use for a glyph not in the font
*.id                            DEF name
*.xml:base                       change base for relative uris
*.xml:lang                       language identifiers
*.xml:space                      preserve white space
missing-glyph.d
missing-glyph.arabic-form
missing-glyph.lang
missing-glyph.horiz-adv-x

animate                        used to animate a single attribute or property over time
*.id                            DEF name
*.xml:base                       change base for relative uris
*.xml:lang                       language identifiers
*.xml:space                      preserve white space
animate.requiredFeatures
animate.requiredExtensions
animate.systemLanguage
animate.onbegin
animate.onend
animate.onrepeat
animate.onload
animate.xlink:type
animate.xlink:href
animate.xlink:role
animate.xlink:arcrole
animate.xlink:title
animate.xlink:show
animate.xlink:actuate
animate.attributeName
animate.attributeType
animate.begin
animate.dur
animate.end
animate.min
animate.max
animate.restart
animate.repeatCount
animate.repeatDur
animate.fill
animate.calcMode
animate.values
animate.keyTimes
animate.keySplines
animate.from
animate.to
animate.by
animate.additive
animate.accumulate

animateColor    ColorInterpolator
    *   DEF name
    *:base    change base for relative urls
    *:lang    language identifiers
    *:space    preserve white space
animateColor.requiredFeatures
animateColor.requiredExtensions
animateColor.systemLanguage
animateColor.onbegin
animateColor.onend
animateColor.onrepeat
animateColor.onload
animateColor.xlink:type
animateColor.xlink:href
animateColor.xlink:role
animateColor.xlink:arcrole
animateColor.xlink:title
animateColor.xlink:show
animateColor.xlink:actuate
animateColor.attributeName
animateColor.attributeType
animateColor.begin
animateColor.end
animateColor.min
animateColor.max
animateColor.restart
animateColor.repeatCount
animateColor.repeatDur
animateColor.fill
animateColor.calcMode
animateColor.values
animateColor.keyTimes
animateColor.keySplines
animateColor.from
animateColor.to
animateColor.by
animateColor.additive
animateColor.accumulate

animateMotion    PositionInterpolator
    *   DEF name
    *:base    change base for relative urls
    *:lang    language identifiers
    *:space    preserve white space
animateMotion.requiredFeatures
animateMotion.requiredExtensions
animateMotion.systemLanguage
animateMotion.onbegin
animateMotion.onend
animateMotion.onrepeat
animateMotion.onload
animateMotion.xlink:type
animateMotion.xlink:href
animateMotion.xlink:role
animateMotion.xlink:arcrole
animateMotion.xlink:title
animateMotion.xlink:show
animateMotion.xlink:actuate
animateMotion.begin
animateMotion.dur
animateMotion.end
animateMotion.min
animateMotion.max
animateMotion.restart
animateMotion.repeatCount
animateMotion.repeatDur
animateMotion.fill
animateMotion.additive
animateMotion.accumulate
animateMotion.values
animateMotion.keyTimes
animateMotion.keySplines
animateMotion.from
animateMotion.to
animateMotion.by
animateMotion.path
animateMotion.keyPoints
animateMotion.rotate
animateMotion.origin

animateTransform.animates a transformation attribute
*.id DEF name
*.xml:base change base for relative urls
*.xml:lang language identifiers
*.xml:space preserve white space
animateTransform.requiredFeatures
animateTransform.requiredExtensions
animateTransform.systemLanguage
animateTransform.onbegin
animateTransform.onend
animateTransform.onload
animateTransform.xlink:type
animateTransform.xlink:href
animateTransform.xlink:role
animateTransform.xlink:arcrole
animateTransform.xlink:title
animateTransform.xlink:show
animateTransform.xlink:actuate
animateTransform.attributeName
animateTransform.attributeType
animateTransform.begin
animateTransform.dur
animateTransform.end
animateTransform.min
animateTransform.max
animateTransform.restart
animateTransform.repeatCount
animateTransform.repeatDur
animateTransform.fill
animateTransform.calcMode
animateTransform.values
animateTransform.keyTimes
animateTransform.keySplines
animateTransform.from
animateTransform.to
animateTransform.by
animateTransform.additive
animateTransform.accumulate
animateTransform.type

mpath use an existing path for animateMotion
*.id DEF name
*.xml:base change base for relative uris
*.xml:lang language identifiers
*.xml:space preserve white space
mpath xlink:type
mpath xlink:href
mpath xlink:role
mpath xlink:arcrole
mpath xlink:title
mpath xlink:show
mpath xlink:actuate

set setting the value of an attribute for a specified duration
*.id DEF name
*.xml:base change base for relative uris
*.xml:lang language identifiers
*.xml:space preserve white space
set requiredFeatures
set requiredExtensions
set systemLanguage
set onbegin
set onend
set onrepeat
set onload
set xlink:type
set xlink:href
set xlink:role
set xlink:arcrole
set xlink:title
set xlink:show
set xlink:actuate
set attributeName
set attributeType
set begin
set dur
set end
set min
set max
set restart
set repeatCount
set repeatDur
set fill
set to
foreignObject  non-SVG element  
*.id      DEF name  
*.xml:base change base for relative urls  
*.xml:lang language identifiers  
*.xml:space preserve white space  
foreignObject.requiredFeatures  
foreignObject.requiredExtensions  
foreignObject.systemLanguage  
foreignObject.graphicsElementEventAttrs  
foreignObject.transform  
foreignObject.x  
foreignObject.y  
foreignObject.width  
foreignObject.height  
foreignObject.content  

color keywords  
multiple transform specifications  

path specification with all those options