SVG and PDF
with Apache Batik and Apache FOP

Jeremias Märki

Topics

- A quick glance at The Apache Software Foundation
- Converting SVG to PDF and other formats
- Apache Batik (SVG implementation)
- Apache FOP (XSL-FO implementation)
- Using SVG in XSL-FO documents
- Lessons learned
- Status and future developments

The Apache Software Foundation (ASF)

- Registered NPO in the USA
- You all know the Apache web server
- Provides an attractive environment to develop Open Source Software
- Companies and programmers get together to develop great software
- Publishes software under a commercially interesting license

The Apache License

- You can download and use for free
- You can modify and redistribute (even for commercial purposes)
- Does not guarantee anything
- Provides legal protection for developers (IP and patent issues etc.)
- One restriction: You must state that you're using software created by the ASF.
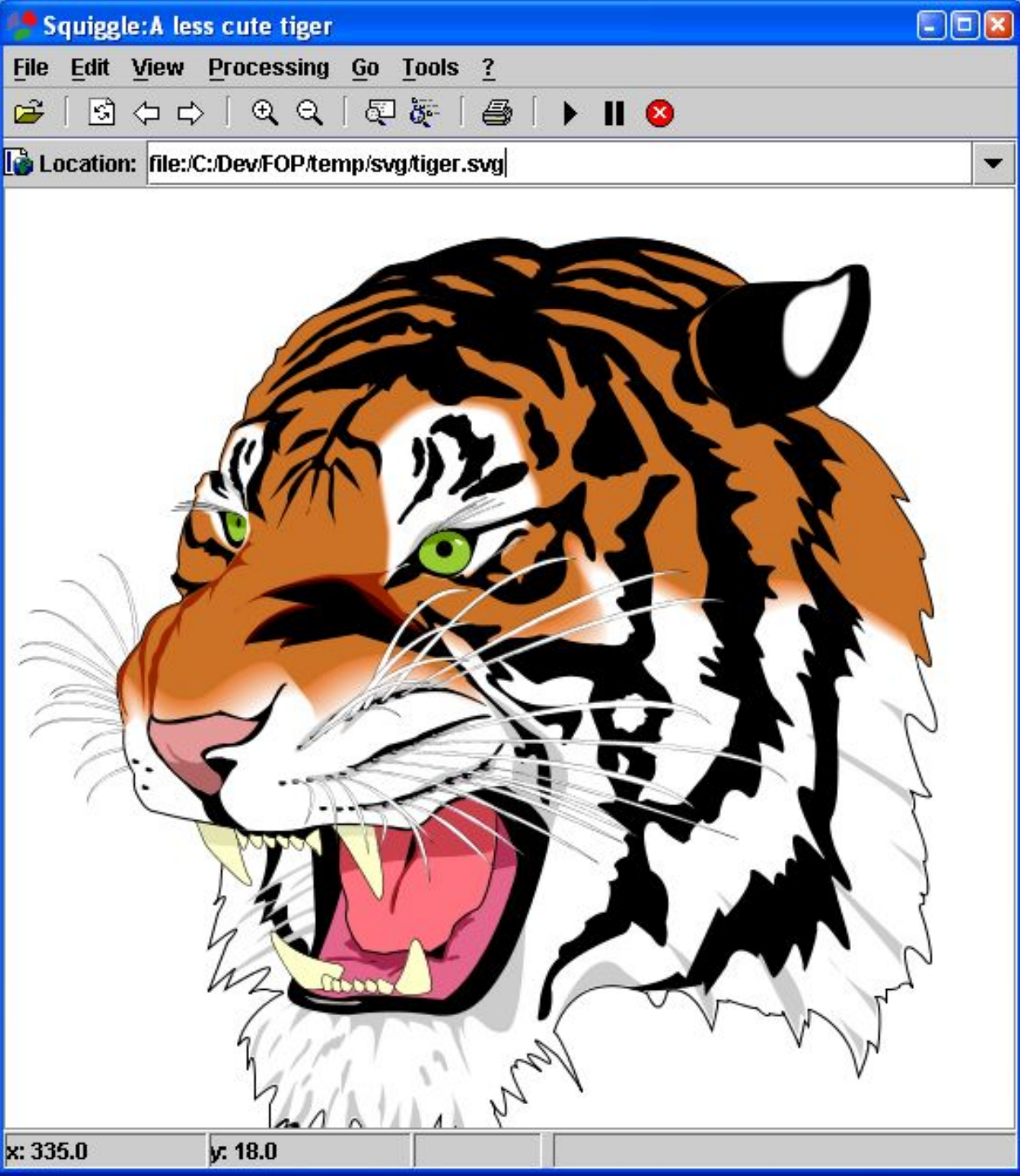- Brand new Apache License 2.0

Roles at the ASF

- Users
  - They use the software, ask questions on the mailing list, help other users, make suggestions
- Developers
  - They modify the software, send in their modifications
- Committers
  - You become a committer through election when you contribute a lot
  - Write access to the code repository
  - They decide on the direction for the project (voting rights)
- PMC members
- ASF members

Apache Batik

- SVG implementation
- Part of the XML project (http://xml.apache.org/batik)
- Current production release: Batik 1.5.1

- Static features: all implemented
- Scripting: all implemented
- Dynamic features: not yet

File   Edit   View   Processing   Go   Tools   ?

Location: file:/C:/Dev/FOP/temp/svg/tiger.svg

x: 335.0   y: 18.0

Batik features

- For end-users: Squiggle, the SVG browser
- Lots of things for developers
- Painting on Java2D for GUIs and printing
- Transcoders convert SVG to other formats
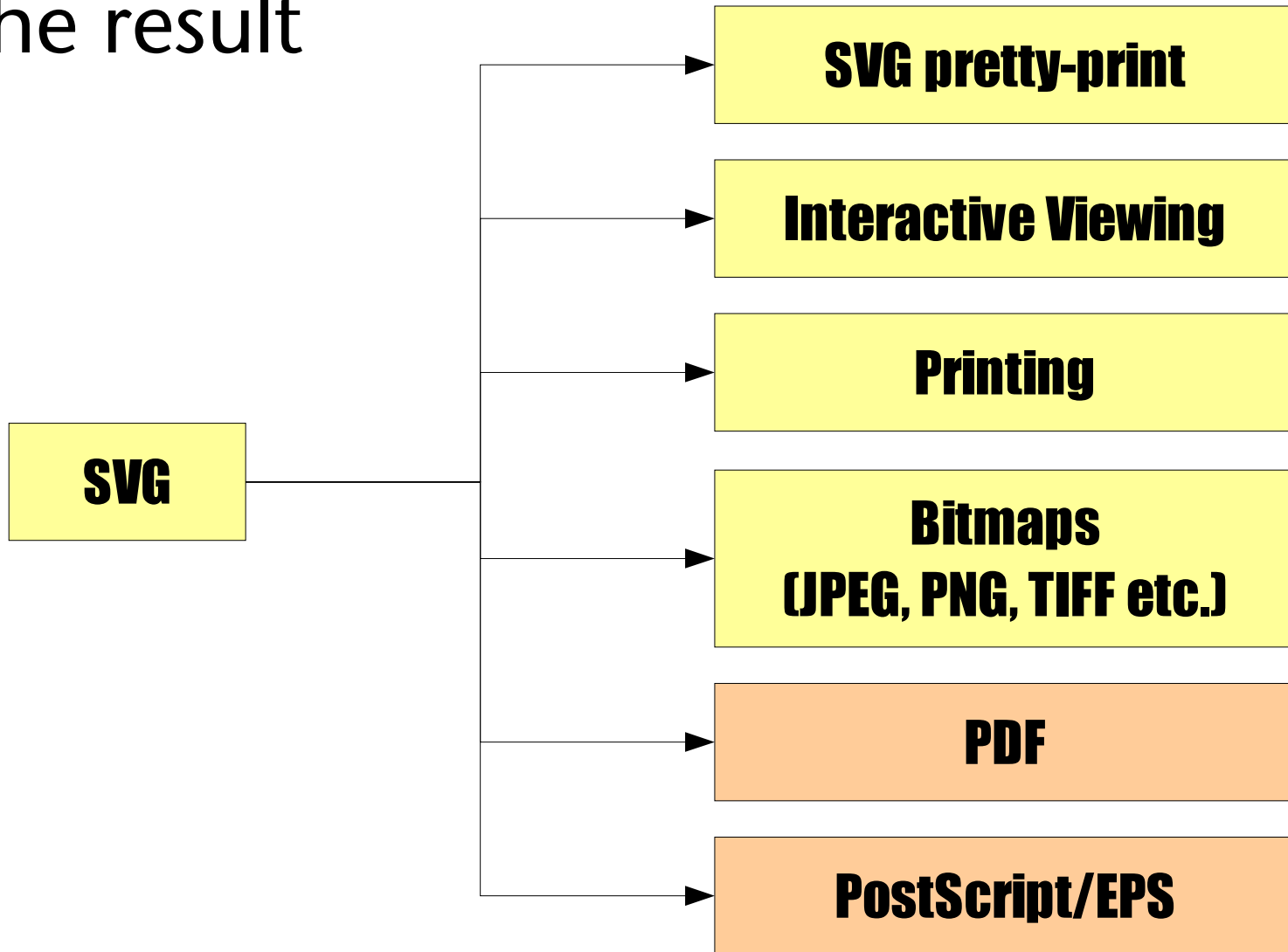
```
                    SVG
   ┌─────────────────┼──────────────────┐
   ▼                 ▼                   ▼             ▼
Bitmaps          Printing        SVG pretty-print
(JPEG, PNG, TIFF etc.)
```
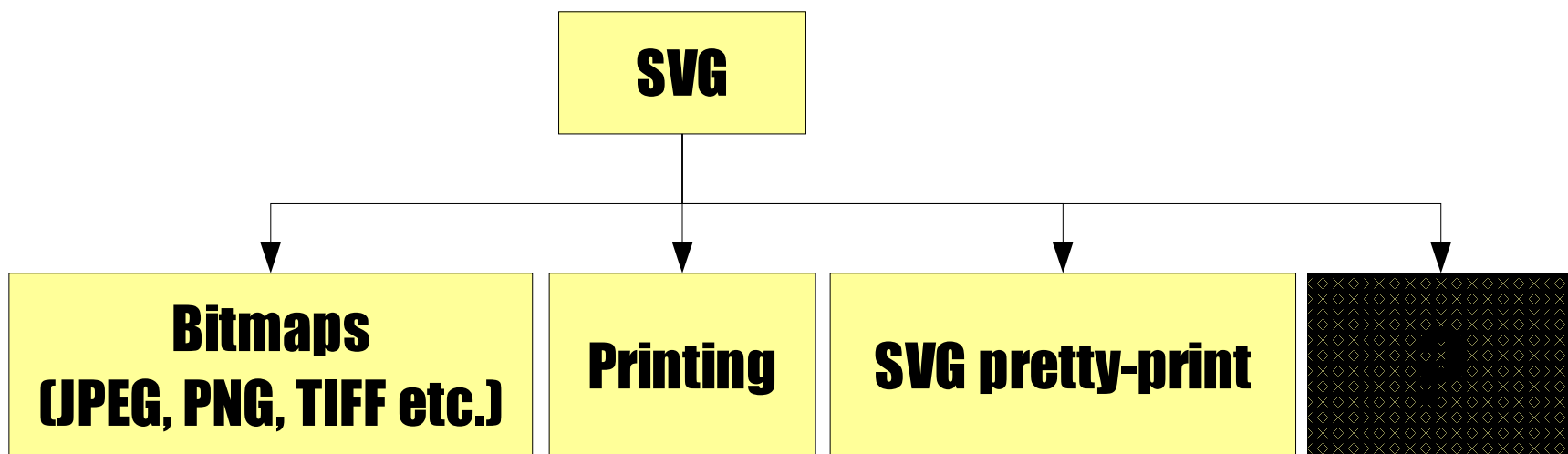
So, where is the PDF?

Apache FOP

- XSL-FO implementation
- Part of the XML project (http://xml.apache.org/fop)
- Current production version: FOP 0.20.5

- Many (but not all) XSL-FO features implemented
- Most important output formats: PDF and PostScript
- Supports SVG through Batik

FOP has additional Transcoders

- FOP needs SVG support, so uses Batik
- We don't want to use bitmaps (Scaling)
- The FOP team developed Batik Transcoders for PDF and PostScript/EPS
- Transcoders use FOP's PDF and PS libraries
- FOP provides a stand-alone package
  as a plug-in to Batik
  (i.e. SVG to PDF/PS conversion without XSL-FO)
- Batik ships with these Transcoders

The result

SVG → SVG pretty-print

SVG → Interactive Viewing

SVG → Printing

SVG → Bitmaps (JPEG, PNG, TIFF etc.)

SVG → PDF

SVG → PostScript/EPS

PDF Transcoder features

- Caveat: Not as mature as the rest of Batik!
- Creates scalable PDF commands
- Works fine for most commonly used constructs
- Some transparency features and patterns
- Support for links

- Test the features you need!

PDF Features inherited from FOP

- PDF version 1.4
- Font embedding for Type 1 and TrueType fonts
- Image support for various formats (JPEG, PNG etc.)
- PDF encryption support (40 bit)

But:

- Only partial ICC support
- Very limited CMYK support
- PDF post-processing tools often needed

PostScript/EPS Transcoder

- Produce either plain PostScript or EPS (Encapsulated PostScript)

- Basically Level 2, but using some Level 3 features

- Good results for simple SVG documents

- Not as advanced as the PDF Transcoder

- No patterns, no bitmap images, yet

- Generic PostScript, no PPD support, yet
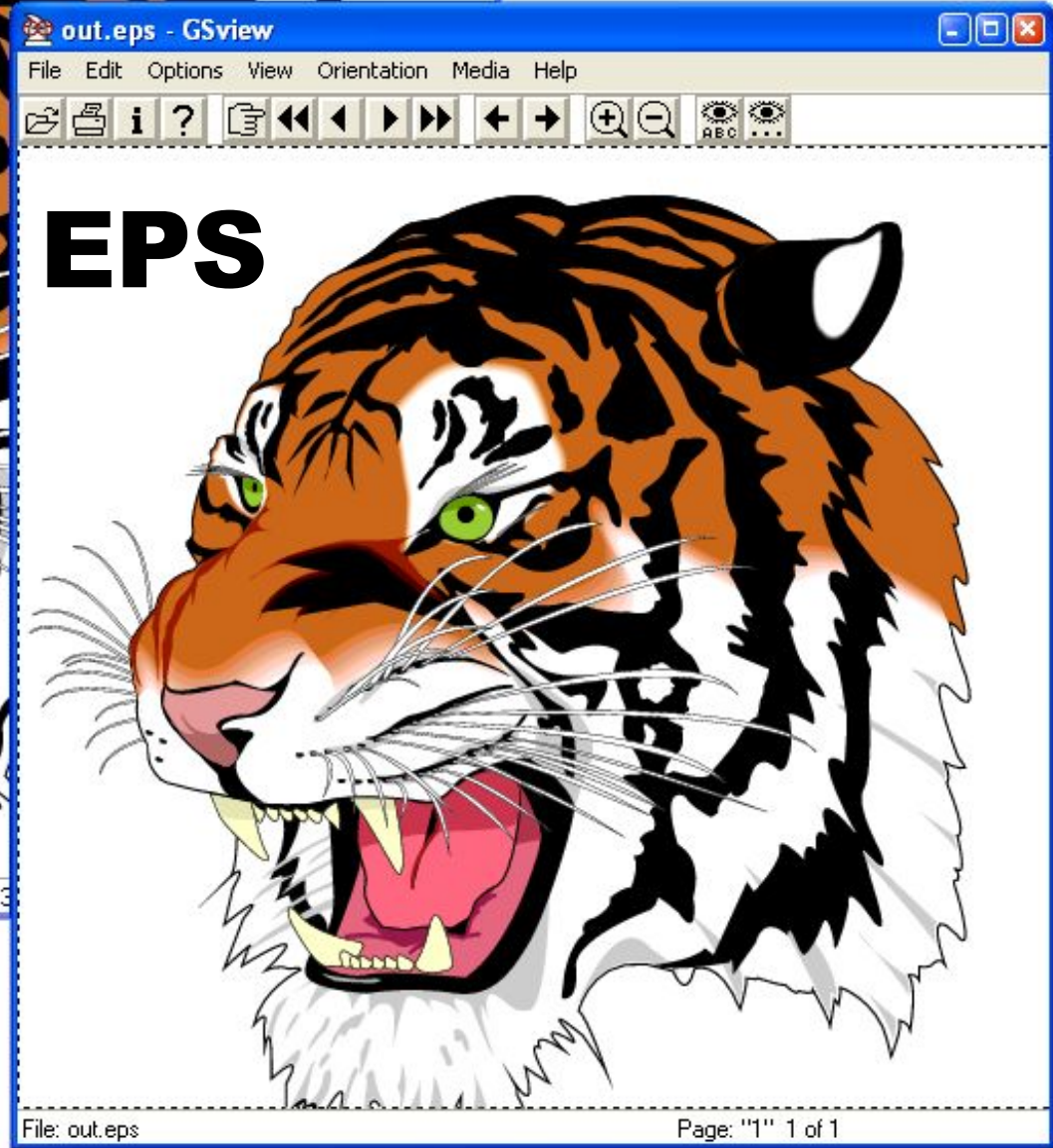
Extending SVG (or XSL-FO)

- Extending SVG provides powerful opportunities
- Both SVG and XSL-FO provide support for "foreign objects"
- Example: Barcodes using the Open Source project "Barcode4J"
- The package transforms a special XML dialect for barcodes into SVG for use inside SVG or XSL-FO.
- A lot of other possibilities: MathML, charts etc. etc.

SVG in XSL-FO

```
<fo:page-sequence master-reference="simpleA4">
  <fo:flow flow-name="xsl-region-body">
    <fo:block font-weigth="bold">
      Logo:
    </fo:block>
    <fo:block>
      <fo:external-graphic src="file:my-logo.svg"/>
    </fo:block>
  </fo:flow>
</fo:page-sequence>
```

```
<fo:instream-foreign-object>
  <svg:svg xmlns:svg="http://www.w3.org/2000/svg"
      width="5cm" height="4cm">
    <svg:g transform="translate(190, 170)">
      ........
    </svg:g>
  </svg:svg>
</fo:instream-foreign-object>
```

Real life usage example #1

- High-volume invoice and insurance policy printing system (personalized documents on laser printers)

- SVG used whenever XSL-FO can't help:
  - Barcodes
  - Logos
  - Form filling

- Generation of PostScript files that are concatenated and post-processed for OMR marks and paper tray selection etc.

- PDF versions for the archive or the customer

Lessons learned

- XSL-FO is very expressive but you need SVG to help out with some specialties.

- Special extensions are easily done with some programming know-how.

- WYSIWYG editors for XSL-FO are not always good enough, manual XSLT still necessary.

- Learn to live with a subset of features theoretically available.

Real-life usage example #2

- Printing of price etiquettes and posters

- Templates designed in Adobe Illustrator (in SVG)

- Apache Batik used to print the generated SVG

- Black and white only, with images and barcodes

- Templates enriched by values from a database

- n-up using separate templates entirely done in SVG

Lessons learned

- XML-based SVG is very flexible, extensible and easy to modify and enrich.

- SVG files can easily be nested.

- As with HTML, interpretation of the SVG standard may differ between implementations.

- You may need to restrict yourself when you're designing templates.

Future for Batik and FOP

- Batik and FOP will grow closer together under a common umbrella: "XML Graphics" (working title)
- PDF and PS libraries will be shared, SVG code in FOP will be more accessible to the Batik team.
- We're looking for more synergies.

- Resources are relatively scarce. Help (in every thinkable flavor) is always welcome. Everyone can participate.

Interesting development

- "SVG Print" working draft

- Directly send SVG to a printer.

- Specifies how to hold multiple pages within one SVG document.

- This moves it closer to PDF.

Conclusions

- SVG is an exciting standard with lots of applications and possibilities.

- I see SVG's strength mainly in complimenting other technologies (SVG in XSL-FO, SVG in PDF, SVG on websites).

- There are good Open Source software packages around which are used in production by many users.

- Do a careful evaluation. Check your requirements.

Conclusions #2

- Possible pitfalls: Color and Interoperability.

- But these new standards make it easier to switch the tool, Open Source or not.

- Editors will still need to become more powerful.

Thank You!

- Feedback? Comments? Suggestions?

Slides available from http://www.seybold365.com

Contact Info

Jeremias Märki, jeremias@maerki.org (Speaker)

David Penfold, penfold@eps-edge.demon.co.uk (Moderator)

Seybold Director of Content, Cynthia Wood, Cynthia.Wood@mlii.com

Seybold VP & General Manager, James Smith, James.Smith@mlii.com

Links

- SVG: http://www.w3.org/Graphics/SVG/

- The Apache Software Foundation: http://www.apache.org

- Apache Batik: http://xml.apache.org/batik

- Apache FOP: http://xml.apache.org/fop

- Apache Cocoon: http://cocoon.apache.org