B.Eng. Final Year Project

Literature Notes

Sam Moore, David Gow
Faculty of Engineering, Computing and Mathematics, University of Western Australia
March 2014

Contents

1 Postscript Language Reference Manual[1]

Adobe's official reference manual for PostScript.

It is big.

2 Portable Document Format Reference Manual[2]

Adobe's official reference for PDF.

It is also big.

3 Portable Document Format (PDF) — Finally...[3]

This is not spectacularly useful, is basically an advertisement for Adobe software.

Intro

- Visual communications has been revolutionised by computing
- BUT there have always been problems in exchanging formats
- Filetypes like text, rich text, IGES, DXF, TIFF, JPEG, GIFF solve problems for particular types of files only
- PDF solves everything for everyone; can include text, images, animation, sound, etc

PDF Features

- Raster Image Process (RIP) — For printing (presumably also displaying on screen)
- Originally needed to convert to PS then RIP, with PS 3 can now RIP directly.
- Reduced filesize due to compression
- Four major applications - Stoy 1999[**?**]
  1. Download files from internet
  2. Files on CDs
  3. Files for outputing to printers
  4. Conventional [commercial scale?] printing
- List of various (Adobe) PDF related software
  - Includes software for PS that converts to/from PDF
  - So PS was obviously pretty popular before PDF
- Can Optimize for screen/printer [not clear how]
- Can compress for size

4 Pixels or Perish [4]

"The art of scientific illustration will have to adapt to the new age of online publishing" And therefore, JavaScript libraries (D^3) are the future.

The point is that we need to change from thinking about documents as paper to thinking of them as pixels. This kind of makes it related to our paper, because it is the same way we are justifying our project. It does mention precision, but doesn't say we need to get more of it.

I get the feeling from this that Web based documents are a whole bunch of completely different design philosophies hacked together with JavaScript.

This paper uses Metaphors a lot. I never met a phor that didn't over extend itself.

Intro

- Drawings/Pictures are ornaments in science but they are not just ornamental
- Processes have changed a lot; eg: photographic plates → digital images
- "we are about to turn the page — if not close the book — on yet another chapter in publishing history." (HO HO HO)
- It would be cool to have animated figures in documents (eg: Population pyramid; changes with time); not just as "supplements"
- In the beginning, there was PostScript, 1970s and 1980s, John Warnock and Charles Geschke, Adobe Systems
- PS is a language for vector graphics; objects are constructed from geometric primitives rather than a discrete array of pixels
- PS is a complete programming language; an image is also a program; can exploit this to control how images are created based on data (eg: Faces)
- PDF is "flattened" PS. No longer programable. Aspires to be "virtual paper".
- But why are we using such powerful computing machines just to emulate sheets paper? (the author asks)

Web based Documents

- HTML, CSS, JavaScript - The Axis of Web Documents
  - HTML - Defines document structure
  - CSS - Defines presentation of elements in document
  - JavaScript - Encodes actions, allows dynamic content (change the HTML/CSS)
- `<canvas>` will let you draw anything (So in principle don't even need all of HTML/CSS)
  - Not device independent
  - "Coordinates can be specified with precision finer than pixel resolution" **(TODO: Investigate this?)**
  - JavaScript operators to draw things on canvas are very similar to the PostScript model
- SVG — Same structure (Document Object Model (DOM)) as HTML
  - "Noun language"
  - Nouns define lines/curves etc, rather than paragraphs/lists
  - Also borrows things from PostScript (eg: line caps and joints)
  - IS device independent, "very high precision" **(TODO: Investigate)**
  - JavaScript can be used to interact with SVG too
- D^3 (Data Driven Documents) - A JavaScript library
  - Idea is to create or modify elements of a DOM document using supplied data
  - `https://github.com/mbostock/d3/wiki`
- We are in a new Golden Age of data visualisation
- Why do we still use PDFs?
  - PDFs are "owned" by the author/reader; you download it, store it, you can print it, etc
  - HTML documents are normally on websites. They are not self contained. They often rely on remote content from other websites (annoying to download the whole document).
- **Conclusion** Someone should open up PDF to accept things like D^3 and other graphics formats (links nicely with [5])
- Also, Harry Potter reference

5 Embedding and Publishing Interactive, 3D Figures in PDF Files[5]

- Linkes well with [4]; I heard you liked figures so I put a figure in your PDF
- Title pretty much summarises it; similar to [4] except these guys actually did something practical

References

[1] Adobe Systems Incorporated. *PostScript Language Reference*. Addison-Wesley Publishing Company, 3rd edition, 1985 - 1999.

[2] Adobe Systems Incorporated. *PDF Reference*. Adobe Systems Incorporated, 6th edition, 2006.

[3] Michael A. Wan-Lee Cheng. Portable document format (pdf) – finally, a universal document exchange technology. *Journal of Technology Studies*, 28(1):59 – 63, 2002.

[4] Brian Hayes. Pixels or perish. *American Scientist*, 100(2):106 – 111, 2012.

[5] David G. Barnes, Michail Vidiassov, Bernhard Ruthensteiner, Christopher J. Fluke, Michelle R. Quayle, and Colin R. McHenry. Embedding and publishing interactive, 3-dimensional, scientific figures in portable document format (pdf) files. *PLoS ONE*, 8(9):1 – 15, 2013.