‘The Camelot Project’

Steven R. Bagley

Introduction

- This lecture is about the birth of PDF
- And it's relation to PostScript
- PDF and Acrobat were originally described in a paper entitled 'The Camelot Project' by John Warnock
- Back in 1991

This lecture is going to deal with some Computer History as well as the documeent stuff

Problem

- Warnock realised that there was no simple way to send a document electronically
- The only reliable solution was to print and post the document
- Or fax it over…
- Pre-internet era

Faxes send images, not documents. Not searchable, scaleable etc high bandwidth and low resolution

Problem

- There were electronic document formats
- But there were lots of them
- And they were incompatible with each other…
- You might also need to transfer images, fonts etc separately…

In 1994, I did work experience in Cripps Computing Centre (as IS was known then) -- the university was running WordPerfect 6. At School at the time we were running Lotus AmiPro. There was also Word, and a host of other formats about

The Camelot Project

- To provide a collection of utilities, applications and system software
- That can capture documents from any application
- Send electronic versions anywhere
- View and print them on any machine

Acrobat, Reader and Distiller

Solution

- PostScript…
- Device-Independent
- Can contain embedded images and fonts
- Easily sent electronically
- All the apps have printer drivers…

People were sending it electronically...
Apps were already producing PS to print, only a matter of redirecting this to a file
No need to get app vendors to rewrite their software

But…

- Not a viable solution
- Requires PostScript compatible printers
- Also PostScript required powerful computers to run
- Typical office computers were under powered to run PostScript

Remember this is the era when your PostScript laser printer was at least as powerful as your computer, probably more so
MEntion Display PostSCript

Spec Comparison

	Mac Plus/ Classic	Mac IIfx	LaserWriter IINTx	LaserWriter IIg
CPU	8MHz 68000	40MHz 68030	16MHz 68020	25MHz 68030
RAM	1MB-4MB	4MB – 128MB	2MB–12MB	5MB–32MB
Price	$999	$9900	—	

All circa 1991
Mac IIfx is the top of the line
FPUs were optional on computers (and often addressed as an IO unit)
PC specs were similar -- although the x86 chip wasn't necessarily as nice as the 680x0 family
Demo CompoScript on Hatari

Speed

- Even with the high-spec printer complex pages could take a while to display
- Operations like page-turning, zooming, even scrolling could be very slow on the typical computer of the day

Speed

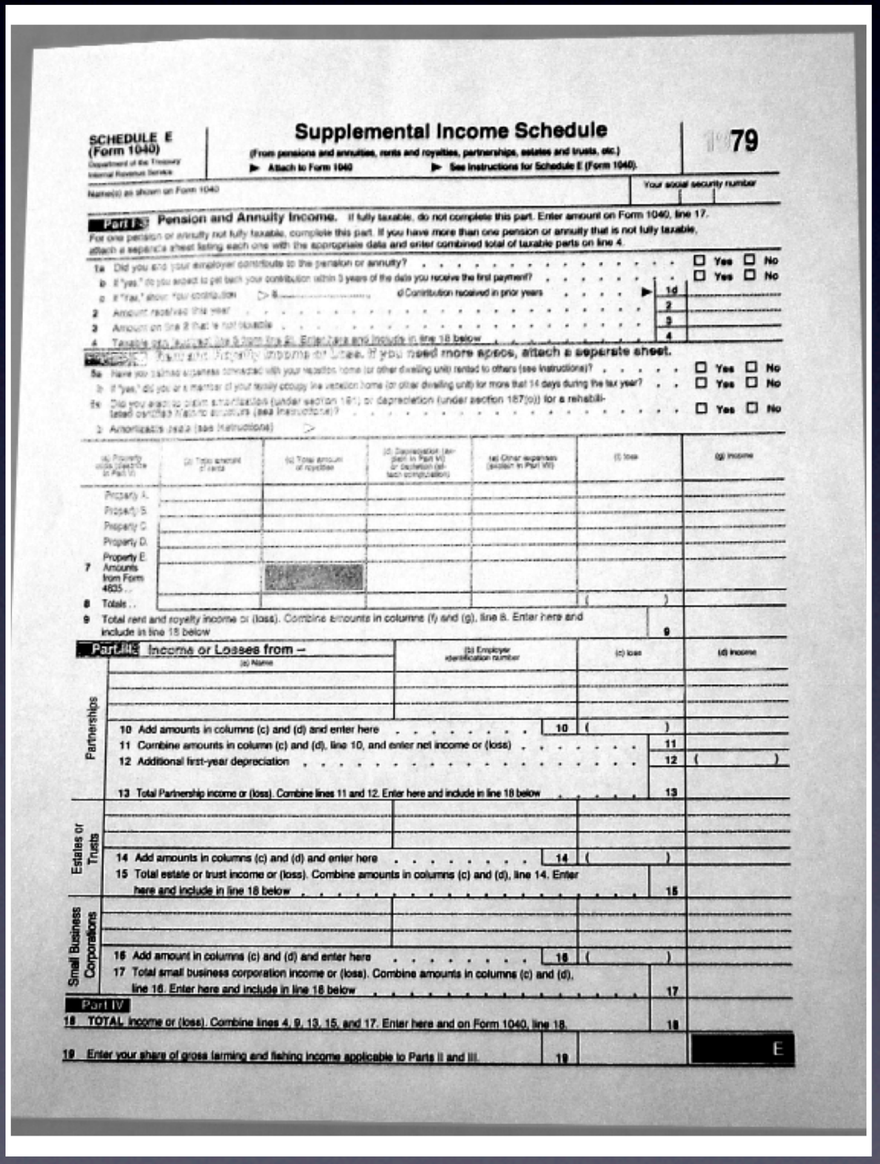
- Two technical approaches to Camelot
- Both dependent on PostScript technology
- First approach was to try and make the interpreter faster
- Had been tried and found to be extremely difficult to do

Divide and Conquer

- Second approach was to divide the problem up
- Each of which on their own would be small enough for the average computer to do
- Needs to be a natural split for the user to use the programs…
- Uses a unique property of PostScript

LaserWriter Launch

- Back in 1984, when Apple was about to launch the LaserWriter
- Adobe generated some sample PS files to demo it including an IRS tax form



SCHEDULE E (Form 1040)
Department of the Treasury
Internal Revenue Service

Supplemental Income Schedule

(From pensions and annuities, rents and royalties, partnerships, estates and trusts, etc.)
▶ Attach to Form 1040 ▶ See Instructions for Schedule E (Form 1040).

79

Name(s) as shown on Form 1040 | Your social security number

Part I Pension and Annuity Income. If fully taxable, do not complete this part. Enter amount on Form 1040, line 17.
For one pension or annuity not fully taxable, complete this part. If you have more than one pension or annuity that is not fully taxable, attach a separate sheet listing each one with the appropriate data and enter combined total of taxable parts on line 4.

1a Did you and your employer contribute to the pension or annuity? ☐ Yes ☐ No
b If "yes," do you expect to get back your contribution within 3 years of the date you receive the first payment? ☐ Yes ☐ No
c If "Yes," show: Your contribution ▷ $ d Contribution received in prior years ▶ 1d
2 Amount received this year 2
3 Amount on line 2 that is not taxable 3
4 Taxable part. Subtract line 3 from line 2. Enter here and include in line 18 below 4

Part II Rent and Royalty Income or Loss. If you need more space, attach a separate sheet.
5a Have you claimed expenses connected with your vacation home (or other dwelling unit) rented to others (see Instructions)? ☐ Yes ☐ No
b If "yes," did you or a member of your family occupy the vacation home (or other dwelling unit) for more that 14 days during the tax year? ☐ Yes ☐ No
6a Did you elect to claim amortization (under section 191) or depreciation (under section 167(o)) for a rehabilitated certified historic structure (see Instructions)? ☐ Yes ☐ No
b Amortizable basis (see Instructions) ▷

(a) Property	(b) Total amount of rents	(c) Total amount of royalties	(d) Depreciation (explain in Part V) or Depletion (attach computation)	(e) Other expenses (explain in Part VI)	(f) loss	(g) income
Property A						
Property B						
Property C						
Property D						
Property E						
7 Amounts from Form 4835						
8 Totals					( )	

9 Total rent and royalty income or (loss). Combine amounts in columns (f) and (g), line 8. Enter here and include in line 18 below 9

Part III Income or Losses from —

	(a) Name	(b) Employer identification number	(c) loss	(d) income
Partnerships				

10 Add amounts in columns (c) and (d) and enter here 10 ( )
11 Combine amounts in column (c) and (d), line 10, and enter net income or (loss) 11
12 Additional first-year depreciation 12 ( )
13 Total Partnership income or (loss). Combine lines 11 and 12. Enter here and include in line 18 below 13

Estates or Trusts
14 Add amounts in columns (c) and (d) and enter here 14 ( )
15 Total estate or trust income or (loss). Combine amounts in columns (c) and (d), line 14. Enter here and include in line 18 below 15

Small Business Corporations
16 Add amount in columns (c) and (d) and enter here 16 ( )
17 Total small business corporation income or (loss). Combine amounts in columns (c) and (d), line 16. Enter here and include in line 18 below 17

Part IV
18 TOTAL income or (loss). Combine lines 4, 9, 13, 15, and 17. Enter here and on Form 1040, line 18. 18
19 Enter your share of gross farming and fishing income applicable to Parts II and III. 19

E

IRS Form PostScript

- Hand-programmed by Warnock
- Used a lot of subroutines to make it easier to program
- Trouble was it took 2mins 45s to print
- Steve Jobs said ‘No, we can’t have any page that takes that long.’

Machine generated PS is a lot easier than generating by hand since it is made by procedures in your program

Flattened PostScript

- Warnock went away and came up with a method that flattened out the PostScript
- Removing everything but the graphics operators
- Called this the ‘graph binder’
- Flattened IRS form only took 22s to display

Binding the graphics operators
Where does the speed up come from?

Rasterizing PostScript

- Rasterizing PostScript basically consists of two parts
  - Drawing the graphics (rasterization)
  - Executing the program

Polygon

```
/poly
{
 1 0 moveto
 /ang 36 def
 10
 {
  ang cos ang sin lineto
  /ang ang 36 add def
 } repeat

} def
```

Here's a simple routine that draws a decagon (10-sided polygon)
Talk through it — highlight how much of it is spent calculating where to go
Particularly stuff like the sin/cos which would be slow

Camelot

- Warnock realised that this same approach could work for PDF
- Split the execution of the PostScript program from its rasterization
- Gives two separate smaller programs which individually could be made to run on a typical computer

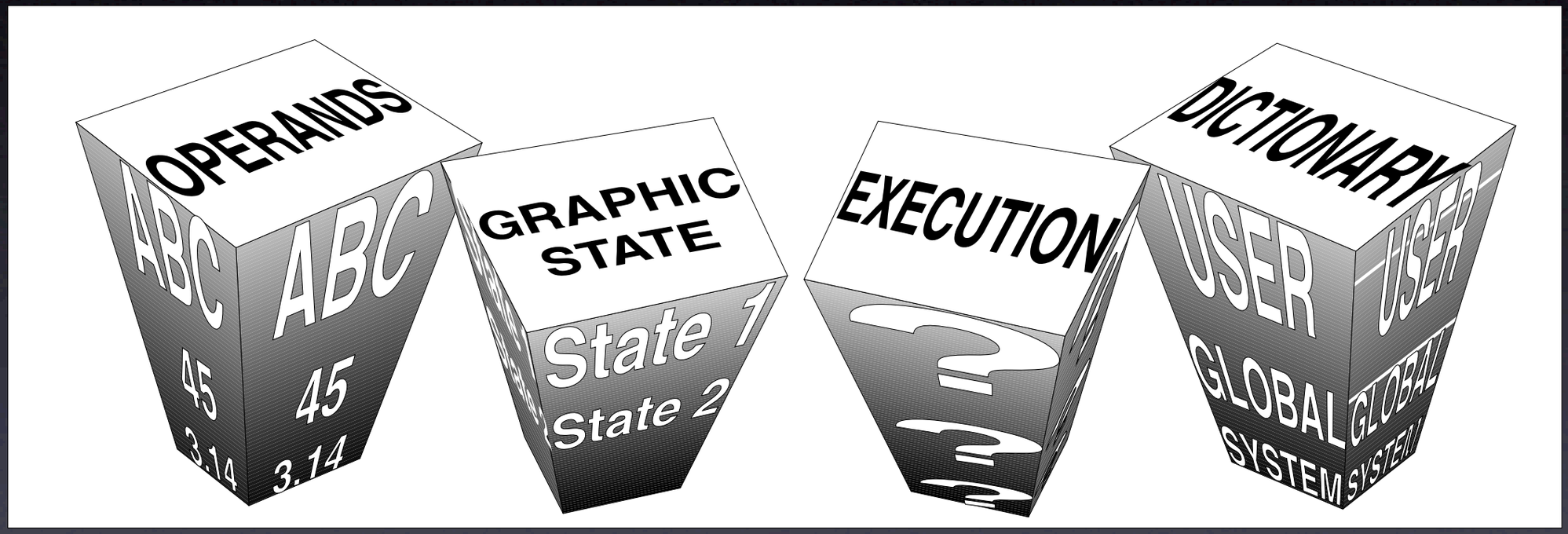
Graph Binder

- How does the graph binder work?
- Relies on the fact that PostScript lets you redefine operators to do something else
- Need to understand how PostScript executes its operators

Operators

- When the PS interpreter executes an operator
- It looks in a dictionary to find the implementation
- Starts with the dictionary at the top of the dictionary stack and works its way down until a match is found

Four Stacks


OPERANDS
ABC
ABC
45
45
3.14
3.14
GRAPHIC STATE
State 1
State 2
EXECUTION
DICTIONARY
USER
USER
GLOBAL
GLOBAL
SYSTEM
SYSTEM

Operators in Dictionary

- The bottom three dictionaries on the stack are `systemdict`, `globaldict` and `userdict`
- Operators are normally defined in `systemdict`
- But if a match is found in a dictionary higher up then it will be executed instead
- Effectively replacing the original operation

Graph Binder

- The Graph Binder uses this trick
- Redefines the graphics operators (`moveto`, `lineto`, `curveto` etc.)
- Instead of performing drawing operations
- The new implementation just print out their parameters and the operator

Show how to implement this with our polygon postscript

Graph Binder

- Executing some PostScript with these redefinitions doesn't draw anything
- It just outputs a new PostScript file to standard out
- If this new file is executed, it'll produce exactly the same graphical output

How could it draw -- we've replaced the drawing operators…
DEMO output as a ps file

Loop unrolling

- This is a classic case of 'loop unrolling' optimization
- Comparisons are one of the slowest things a computer can do
- Straight line code always runs faster
- So by removing the comparisons we get a much faster program

Interchange PostScript

- Warnock realised they could define *Interchange PostScript* containing only the graphics and imaging operators
- Once defined an *IPS binder* could take an input PostScript file and produce an IPS file
- A separate viewer app could then view the IPS files to read the document

Problem split in two

IPS Binder

- Version of the PostScript interpreter
- With the equivalent of Graph Binder built-in
- Developed by Doug Brotz
- Equivalent to what is now known as *Adobe Distiller*

IPS Viewer

- Prototype Viewer created that could display the IPS files
- By a small team
- The result was remarkably fast for the machines of the time
- Equivalent to Adobe Reader or Acrobat now…

COS

- And there we have the basis of PDF
- The one thing that remained was a file format to gather all these things together
- This was designed by Peter Hibbard and called the *Carousel Object System*

Carousel was the original name for Acrobat…