MetaPost: how we adapt

1 Development Stage

I we started with simple usage (logos) and PostScript output

II then we moved on to conversion to pdf using TEX macro solution

III this has the advantage that fonts are handled by TEX

IV for a long time this was a generic solution (later became the MkII variant)

2 Development Stage

I we added some extensions (transparency, cmyk, etc) and MetaFun showed up

II that extension mechanism uses special colors as signals

III we always collected btex ... etex in order to speed up processing

IV in addition we added textext and similar features

V communication between MetaFun and ConTEXt became more advanced over time

3 Development Stage

I when LuaTEX showed up a substitution based lua converter was written

II later when lpeg came around an experimental lpeg converter showed up

III some changes were made to textext processing and run management

4 Development Stage

I the arrival of mplib had rather big consequences

II integration of MetaFun became less of a runtime burden

III a more definitive lpeg converter was written

IV text handling was kept internal (but still needs two passes)

5 Development Stage

I some extensions were changed to use the new pre/postscripts functionality

II the lpeg converter was adapted accordingly

III multiple (independent) METAPOST instances were now supported

IV the chemical code was overhauled and moved to the core

6 Development Stage

I by now all extensions use pre/postscripts

II this made it easier to add more extend functionality

III again the lpeg converter was adapted (simplified)

IV it became possible to do some color trickery with text

V but (till now) color spaces are more complex (mp has a mixed model)

VI so we might move towards a slightly different approach

VII a couple of helpers were added for Mojca (some more will follow)

7 Development Stage

I we're now splitting the code in MpII and MpIV code bases

II from now on the focus will be on MpIV

III the (rather old) MetaFun code will be cleaned up

IV where possible namespaces will be added

V as MetaFun us loaded runtime we see an impact on startup time (quite some files)

VI so maybe we will use a packed and/or zipped pseudo format file for faster loading

8 Development Stage

I an overhaul of the flowchart code is on the agenda

II we also want to finish (and cleanup) the chemical related code

III simple data/graphics helpers will be provided (graph replacement)

IV and of course we keep moving on (who knows what METAPOST 2 will bring us)

V some examples: tests/mkiv/metapost/plugins-*