luaTEX

going beta

TUG 2007

Hans Hagen

Hartmut Henkel

Taco Hoekwater

a bit of history

bringing Lua into TEX is inspired by the SciTE editor

being challenged by Hans, Hartmut built Lua into TEX

and it was not hard to talk Taco
into what would become luaTEX

at that time Oriental TEX (by Idris)
needed a replacement for Aleph

when funding for code development
became an option, the project got a boost

Taco does the coding, Hans deals with the
Lua side, together we Skype up the specs

we are convinced this way of extending
TEX is going to work out well

a few of the objectives

internal quantities will become 21–32 bit

instead of 8 bit input UTF-8 has to be used

some optimizations are going to be removed

flexible support for OpenType fonts is a necessity

the bidirectional typesetting of Aleph will replace eTEX's

Lua callbacks will open up most
of TEX internals and procedures

we need some extensions to TEX (for instance math)

MetaPost will be integrated into TEX

the way we work

stepwise development is planned (we have an agenda)

development occasionally peaks when critical
component are done (unstable versions)

we explore ideas, write code, test
in real situations, rewrite, . . .

we try to keep up with documentation and
development notes (soon we start working on a book)

how it all fits in

pdfTeX: 1++ will be frozen (only debugged)

luaTeX: is de facto pdfTeX 2++

TeX-Gyre: this project will provide the fonts

mplib: MetaPost becomes an embedded library

MkIV: the next generation ConTeXt (our testbed)

where does Lua show up

users can directly call the scripting engine

there is an independent binary (texlua)

developers can extend and/or replace existing functionality

new functionality is exclusively written in Lua

for as much as needed, helper code
is written in C (e.g. font loaders)

we have:

the \directlua command

catcodes management

possibilities to overload (file) io

tfm and OpenType font loaders

hooks into the tokenizer

node handlers and node attributes

and more (to come)

but:

no precooked solutions

where do we stand

we plan to open up the paragraph and page builder

we will open up and extend math support

error handling will become scriptable

in the process existing code
will be cleaned up cq. replaced

eventually the TeX kernel
components will be rewritten in C

in the end, Lua code will connect all TeX components

how about the future