Howling
to the moon
A few observations

— Occasionally we reach the boundaries of TeX and programming then becomes kind of cumbersome.
— There is much macro code around and there are many documents that depend on TeX's longliveness.
— Quite some applications demand more advanced programming features than TeX can (conveniently) provide.
— It's mostly some of the subsystems that we'd like to extend and/or replace.
— Access to the internals, exiting features and possible extensions makes it possible to delegate activities to other programs.
— Every now and then I need a new challenge.
How it started

— I use SciTE as main editor for documents and programs. This editor is built on top of the Scintilla framework.
— Being mainly a program editor it lacks some features that you need for text editing and processing.
— However . . . it has a scripting engine that gives access to the user interface and the editing component.
— This scripting language happens to be Lua.
— For a while I was hesitant to use yet another programming language.
— Where the Perl, Python and (my favourite) Ruby languages are heavy--weight languages, Lua is a simple but nevertheless powerful language.

— P, P and R come with a constantly growing amount of libraries and users expect the whole lot to be present.

— Lua can interface to libraries but its main application is to be embedded in an application and present the user with an interface to the parent application.

— Lua has a small footprint; it add some 50–100K to the parent binary, but one can add additional functionality, for instance support for sockets.

— The more I looked into it, the more I liked it.

— And, after positive experiences with integrating METAPOST, integrating Lua didn’t seem that strange.
And so . . .

— I managed to trick Hartmut into implementing a \texttt{lua} command. (We share a passion for experiments in \TeX{} and he knows how to implement them.)

— I got back a (linux) binary within a few days.

— After we had this proof of concept, it was time to involve Taco, who knows more about \TeX{} than I'll ever know.

— Another few days later we had the first interface to some of \TeX{}'s internals.

— And since we recognized the potential we decided to continue this effort and the \texttt{LuaT\TeX{}} team was there: Hartmut Henkel, Taco Hoekwater & Hans Hagen.

— We will discuss, meet, develop and finally present a working and stable version, probably around Euro\TeX{} and/or TUG2006.

— The binary will start as a fork of pdf\TeX{}, and in the end may replace pdf\TeX{} since Thanh likes the idea.
Potential

— Remove some of the fuzzy parts of pdfTEX (candidates for removal: enctex, mltex).
— Replace some subsystems and/or prototype new subsystems.
— Provide an interface to plugins, like for instance paragraph building components like the ones Karel Skoupý has demonstrated. (We had already discussed this possibility at EuroTEX.)
— Provide a playground for all kind of features, like for instance language/script specific font handlers and paragraph builders for instance for Chinese.
— Replace (or extend) parts of macro packages by alternatives in a higher level language (ConTEXt 4).
— Add functionality to macro packages that until now was beyond reason to implement.
— And this is just the start . . .