TEX contra TEX–remarks by Chris Rowley for BachoTeX 2008

1. Side effects
   In pure TEX, expansion is almost completely 'side-effect free' ie the only thing that changes as a result of the expansion mechanism is the token list being processed.
   In luaTEX, some lua 'script' is called by expanding a suitable TEX command, which makes the design change in 1. even more powerful (=frightening, for me): that lua script is intended to change the TEX machine: ie to be entirely side-effects.
   Put another way: macro expansion is functional programming; lua (at least in its effects on the TEX machine) is procedural.
   Both these programming paradigms have their uses and there is nothing immoral or bad in using functional and procedural programming in the same system but too intertwine them in this way and to turn part of TEX that was purely functional into the complete opposite seems in principle dangerous to me.
   More pragmatically, the power of being able to probe and change almost anything in such a complex monstrosity (oops, monolith) as the TEX machine in this unstructured and uncontrolled way is dangerous (at least in anyone except Hans' hands).
   More personally, when someone as clever and demanding as David Kastrup wants to combine the power of luaTEX's procedural methods with the fragile but enormous structure of LaTEX's many layers of functional programming and declarative intercaes, this could put unbearable pressures on both system's (the people that is).
2. Mutiple MVLs please!
   I believe that luaTEX only supports one TEX machine (and hence only one Main Vertical List). This makes it unsuitable for high quality typesetting of a document that is non-linear in either its logical form (eg in a DOM, as most parsed documents now are) or its visual form. Even adjusting paragraphs to fit into pages in something as simple as a chapter of a novel would be easier if all their possible formattings could easily be determined in parallel by running several TEX machines (with luaTEX hooks to retrieve information about the formattings, of course).
3. The Maths World
   The Maths World needs a Standard LaTEX (christened LoTEX by two of us last summner): this is a formally standardised language that will be a subset of 'LaTEX+packages-swhat it can do used-for-mathsmode-stuff'.
   (The name LoTEX was the least bad choice! The idea being that it is the Lowest Common Denominator.)
   This is not a standard that will ever be mandated (although it may be in for example: math-as-LaTEX-in-HTML5.0).
   It is needed as a reference so that when a (math) application says it 'supports LaTEX-encoded maths' this can be made more precise by saying:
   it supports
   LoTEX plus ... except ....
   It would be good to have official TUG support for this idea and essential to get TUG to approve it and advertise it widely once it exists.

4. Algorithm liberation
   The world loves TEX for certain specific typesetting algorithms it contains (not all of them but many of them) but hates a lot of the old baggage that smothers these, and especially hates its command-based stucture and interfaces (both human and APIs, ... what APIs?). The things they love (even if they do not know what these are) are, at the programming level, the basic data structures and algorithms of which the two most important are:
   the paragraph-maker
   the math formatter
   Of course, we know that these depend on some more basic stuff such as the box-glue methods, hyphenation, etc.
   Therefore what the world needs is not a monolithic TEX, even with luaTEX's tentacles reaching into it.
   The world needs TEX's mid-level formatting methods/algorithms liberated from the monolith and available as libraries to build into today's more advanced document processors, both generic and specialised, using declarative mark-up and DOM-like structures.
   I think that TUG should be putting a lot of effort into this one. An immediate application of liberating the maths formatting would be to get maths on the web looking better than either Firefox or MathPlayer make it appear natively (but not as good as Murray Sargent's RichEdit, of course :-)

   Let's see what happens in 10 years time.

   If you want me to defend something I am happy to defend XeTEX and Jonathan's expertise and philosophy against allcomers.