

One by one the guests arrive

Kees van der Laan

Hunzeweg 57,
9893 PB Garnwerd, The Netherlands
cgl@rc.service.rug.nl

Abstract

A plea is made for writing macros in plain T_EX sufficiently documented to be used with all flavours of T_EX.

1 Introduction

This note emerged from a request of Sebastian Rahtz triggered by my message which I passed along with my public appraisal for the 100 (L^A)T_EX FAQs.

This plea, this shout, hopes to awake the notion that we are all better off if we write macro *software* in the lowest common set of all T_EX flavours. At least it might initiate a discussion because I'm realistic enough that not all involved share my views.

Of course I know that reality is more complicated, and that a right balance is the best we can opt for, so let us go for that.

2 Why

would a L^AT_EX devotee ask? Do you have concrete arguments? Well, from my own experience I can say that there was a time I needed typesetting number ranges. Only the L^AT_EX style of Donald Arsenuau was available. But, what I needed was a few macros to cooperate with plain, so I had to write one of my own,¹ which by the way emerged as a much, much more compact suite. After all the need has faded away because I tackled the handling of bibliography references more fundamental. The point is that it would have been better if there had been a kernel independent from the higher layer which I could have taken over. The interface towards the higher level, or let us say the user interface, should better be built on top. The paradigm in this example is the awareness of CISO, as analogy of FIFO, meaning Collective In and Smallest Out, which solves the problem.

That this approach is beneficial in software engineering in general has been proven by the various numerical software program libraries, which have the basic material written in the lowest language feasible, FORTRAN, allowing stability, optimization of the code, and confidence in use. Similar, I remember the PDE (partial differential equation) packages which use common basic algorithms, but differ in the jargon at the user level. I hope that the macro/package/module writers have a feeling for the savings of the costs which can be gained over time, by this attitude. As a volunteer organization one could shrug it off and say I don't care, costs are not relevant. Then there is

still another nasty guy lurking around the corner that the (All)T_EX community like various sects will fall apart, will fragment. To continue the tune

```
And no one knows where the night is going
And no one knows why the wine is flowing
O love, I need you, I need you, I need you
I need you now
```

Another example is how to provide for headings? The answer is that I don't care so much about heading macros because the common part is so negligible, while it is highly intertwined with the user interface. But—there is always a but—I for one am strongly in favour of starting from two-part macros, which should perform the essential functionalities whatever you may wish, and build all the ornamenta—i.e., the user-interfaces, possibly with less functionality—on top. This approach obeys the *separations of concerns* principle, and pays off in maintenance, if not that it spreads more easily.² To give you an idea of how I did it basically in blue.tex

```
\def\beginhead{<the required functionality>}
\def\endhead{<the required finishing off>}
%with as one-part on top
\def\head#{\bgroup\beginhead
\aftergroup\endhead
\afterassignment\ignorewhitespace
\let\dummy=}
%or the tribute to mammac
\def\bluehead#1\par{\beginhead#1\endhead}
```

The last tribute lost the processing on-the-fly functionality, but most of the time I don't need that, at the expense of simpler markup. But the latter is a matter of taste, I know.

If people like a L^AT_EX-flavoured header just go ahead and add it. The fundamental functionalities have been provided already, just a user interface has to be provided as variant.

3 Conclusion

The point I'm trying to make is that we are all better off when complex fundamental parts will be programmed in plain, perhaps after it has proven to be a fundamental point. To end Cohen's song

```
The guests are coming through
The open-hearted many
The broken-hearted few
```

¹Who cares? It is estimated that 80% or more of the software is continuously rewritten, that is a fact. My reply is that we can do much better, and we should if we opt for the best.

²Forgive me this joke, with L^AT_EX widespread.