

D3: The re-use of re-use

Les Hatton, lesh@oakcomp.co.uk

Nov 1996

Hi. I thought I would talk a little about re-use and its effects on reliability this month. I was saving this for later, but I have just read an article entitled "Objects are dead, long live the component ...". The essence of the story was that objects have been and gone as they have not delivered on their promise of re-use. I have always resented the notion of code re-use being purloined by OO and component practitioners, whatever they are, so at the risk of appearing as the quintessentially English phrase so aptly puts it, an old fart, let me tell you a little story.

In the early 1970's, I was a poor but happy research student working on the dynamical structure of tornadoes. Geophysical fluid dynamics is notorious for leading to truly horrible sets of equations to solve, and yet my programs for solving these and displaying them were quite simple and amounted to only a few hundred lines of code rather than the hundreds of thousands we tend to go in for nowadays. Why were my codes so small ? The answer is very simple, I like countless other contemporaries used something called re-use as clearly described by Alan Turing over 40 years ago. With this amazing concept, you would actually use code that other people expert in a particular area had written ! If that weren't enough, there were even libraries of these called, let me see if I remember, oh yes, component libraries - I used the venerable and distinguished NAG libraries and was very glad to do so indeed. In these libraries were many wonders including such sybaritic delights as components which solved coupled non-linear differential equations which occurred unpleasantly often in my field. The reason these libraries were used was because if you actually wanted to achieve anything in your 3 year's research, the last thing you did was sit down, write and debug components which solved these mathematical horrors unless you were a) masochistic, b) a numerical analyst, (see (a)), or c) completely raving, (see (a) and (b)). When we wanted to draw pictures, we used something called graphics libraries, which also contained code we hadn't written. The language we used was called Fortran which contained an amazing feature - it allowed you to call subroutines, and these could be written and pre-compiled by somebody else. Heady stuff eh ?

I realise that this is pretty high-tech, so much so that we seem to have a desperate need to re-invent it all again, but this time with so many bells, whistles and toys that it is near unusable. In scientific computation, re-use ratios of 90% or greater were regularly achieved otherwise you would go balmy. I'm afraid I can only hoot with derision that objects are failing because the re-use ratio is so

low according to the aforesaid article, (and other sources I have seen). Its really very simple. You can only re-use something if you can understand it. Sooner or later, I suppose we will get the point, (again). Season's greetings.