D3: Software - dross ex machina

Les Hatton, lesh@oakcomp.co.uk

Nov 1995

Last month I promised to widen the net a little from PC's, whose software reliability leaves a lot to be desired. Incidentally, have you noticed in the Unix v. nearly everything else wars of the past few years, that nobody ever mentions reliability. Whilst the Mac OS, Windows NT, '95 and so on exhibit considerable evidence of unreliability, Unix almost never crashes. I can remember one occasion across multiple machines in several years in my own company. This is obviously unimportant to many observers who seem to prefer a Ferrari with no engine to a Land Rover, but I digress.

The real issue I wanted to address here is the massive and alarming growth of software in consumer electronics in recent years. I left you last month with a question - how would you like your favourite piece of consumer electronics to have the same software reliability as a PC ? The problem to come is based on two issues. First, according to very prestigious sources such as NASA, the overall fault density or faults per line of code has been roughly constant in the last 15 years, showing only marginal improvement. Second, the amount of software in consumer electronic devices is doubling around every 18-24 months. For example, there is around 150,000 lines of C in the next generation television and it is obviously only a matter of time before we see the electric toothbrush with a C compiler. Put these together and you are left with the inevitable conclusion that the number of software faults in consumer electronics will approximately double every 18-24 months. Oh dear.

To see the excitement that this can cause, picture yourself on a package tour plane like flight G-VAEL, a brand spanking new fly-and-flush-toilets-by-wire Airbus A340, which eventually landed at Heathrow in Sept 1994. First of all, its software calculated the amount of fuel remaining incorrectly, leading to a full emergency landing during which both control screens went blank, being replaced by reassuring "Please Wait ..." messages. In a further fit of software pique, the plane turned right when instructed to turn left and also acquired a 9 degree plummet rather than a more leisurely 3 degree descent.

Some of these faults exhibited classic behaviour. For example, the uncommanded right turn occurred on a heading of exactly 180 degrees, a very well known type of software fault resulting from a boundary error which I'll discuss later. The A340 is by no means alone as the problem is quite simply this: we are not good enough at software yet for certain levels of system criticality. I hope that we don't do anything too silly before we get there. Next month, I'll start talking about beliefs we hold dear which are not supported by recent data starting with languages. Prepare yourself for a few surprises.