# Computer still says …

*Les Hatton and Michiel van Genuchten*

It used to be a standing joke with computer systems when questioning their frequently suspicious outputs.  You might call a computer support line to complain about a strange telephone bill, only to be told that the bill is correct because "Computer says …". even if the bill was unusually large.  Of course the effects of software errors are limited only by the imagination and "unusually" large can really mean just that, such as happened with the unfortunate French woman in 2012 who received a telephone bill for EUR 11.8 quadrillion[1].  If you are unaccustomed to numbers like this in your telephone bill, it is hardly surprising as it is around $13,000,000,000,000,000 or just under 5,000 times the GDP of France for that year; some phone bill.  To be fair to the lady's phone company *Bouygues Telecom*, they did offer to allow her to pay off the bill in instalments, a most generous offer, although they didn't say how many instalments might be necessary[2].

Further to be fair to *Bouygues Telecom,* they are not alone.  In April 2006, a Malaysian man received one for just $218 trillion[3], or $218,000,000,000,000.  In his case, his phone provider *Telekom Malaysia* generously gave him 10 days to pay or face legal proceedings, a situation which the gentleman concerned is alleged to have said "I can't wait to face".  We don't blame him because we can laugh at such cases; in fact we must.  They are truly ridiculous and even the bureaucrats in the companys concerned will have eventually realised this, although we may be being overly generous, given that they issued them in the first place.  However, this conceals a dangerous precedent.  What if the bill had still been enormous, just as wrong, but crucially (just) feasible, say $10,000?   How much confidence would we have that the company issuing this bill would back down?  I think we all know the answer to that.

What this all reduces to is that in all organisations, it is convenient and far too likely that the organisation will hide behind the mantra "Computer says …".  In other words, "we" are right and you are wrong because "we" either can't or more likely won't countenance the fact that the "computer" could be wrong, for legal reasons, pride, pure ignorance or whatever. A complicating factor is that software solutions nowadays consist of many layers typically provided by multiple vendors. One of the early Impact columns was about banking systems. The authors already explained that in the past there would typically be 3 companies involved in a banking systems: the company supplying the operating system, another for the database and a 3rd one for the banking applications. That meant that in case of an outage, technical experts from the three companies would meet, roll up their sleeves, fix the problem and later sort out who caused the problem. Now there may be many more companies contributing to the software stack. An initial meeting when an outage hits may be 10 technical experts accompanied by 10 lawyers who stare at each other not willing to admit any guilt. By the way, having the complete software stack being developed by one company with a market cap that would allow them to supply all the banks in the country is no solution either.

[1] https://www.bbc.co.uk/news/world-europe-19908095

[2] Hardly surprising.  We know you are desperate to know so at $1,000 a month, it would take about 50 times the current lifetime of the universe taking bureaucratic innumeracy to new and giddy depths.  No mention was made of accumulating interest payments.

[3] https://www.nbcnews.com/id/wbna12247590

But we and the readership of *IEEE Software* are software engineers and when we hear some helpless bureaucrat quote “Computer says …” we know that they are really saying something like *“The computer software has output some data and we have no idea how it produced this but we know it must be right, because, well, we just know and the last thing we want to do is attract any publicity/liability or whatever, so we will just hunker down and hope they pay up and it goes away.”* However, readers of this column will also know that faults in computer software are all too common and remain so in spite of Herculean efforts to the contrary (Hatton 2024). For example, thanks to the pioneering work of researchers such as Ed Adams at IBM, (Adams 1984), we also know that such faults can lay hidden for years before making their influence felt or even worse, have made their influence felt continuously without anybody realising. Both of these scenarios are painfully frequent.

## The UK Post Office fiasco

Of course, the reason we are re-visiting this is that it is still going on as evidenced by the 25 year saga of the UK Post Office v. its own postmasters[4]. They are not alone in this kind of behaviour but this is an exemplary study of cover-ups, bullying and incompetence of literally staggering proportions which have collectively ruined the lives and reputations of many innocent people. It should never have happened and when it finally came out (and at the time of writing is still coming out), should have led to prompt and generous reparations. It hasn’t as yet, to the shame of the perpetrators. This saga is “Computer says …” taken to its most extreme form of toxicity.

### *The Horizon sub post-office software*

In 1999, a new piece of software known as *Horizon*, written by sub-contractors *Fujitsu*, was installed in all the sub post-offices in the UK. This was supposed to take over all accounting and stocktaking processes which are many and various. It was obvious from the earliest days to sub-postmasters that the new system was buggy. It will come as no surprise to the readers of this column that new systems are buggy. However, this system was reporting significant shortfalls in received money of the order of many thousands of pounds. The Post Office would not countenance the possibility of errors in this software so immediately blamed the sub-postmasters threatening legal action and the investigation and pursuit of any shortfalls. Needless to say, this brought chaos into the lives of the victims. Over the next few years, marriages broke down, many made up the shortfall out of their pockets causing financial ruin because the Post Office insisted they were personally responsible for any “unexplained losses”, others faced bankruptcy and some took their own lives. All in all, more than 900 sub-postmasters and sub-postmistresses were prosecuted for stealing money by the Post Office and another 300 or so by the Crown Prosecution Service.

The scale of this is truly breathtaking. The Criminal Cases Review Commission in the UK stated that the scandal was “the most widespread miscarriage of justice it had ever seen, and represents the biggest single series of wrongful convictions in British legal history.”

There is nothing even remotely funny about “Computer says …”. It would be nice to report that eventually, the truth would come out and people did see sense but this is not what happened at all. The Post Office continued to deny any possibility of errors in the Horizon system and it began to cave in only when a program aired on the independent television channel ITV in the UK as late as

[4] https://www.postofficehorizoninquiry.org.uk/, accessed 30-May-2024.

2024[5] detailing the efforts of victims to get some kind of justice and the true scale of this sorry tale was laid before the British public overwhelming the Post Office by a tsunami of public opinion. Things are at last happening but still far too slowly for many – as of mid January 2024, only 95 convictions had been overturned, mostly for technical legal reasons and compensation has been very slow in coming for less palatable reasons. Only in late May, were all convictions finally quashed by Act of Parliament immediately Parliament was dissolved in preparation for the upcoming UK General Election.

This whole sorry tale is the latest and indeed by far the most egregious example of "Computer says …", but there is a long and sordid history of such instances.

### *The Chinook Air Disaster 1994*

Reluctance to admit the possibility of computer software failing is both widespread and corrosive. In this tragic example, 29 people were killed when a Chinook Mk 2 helicopter crashed in cloud on the Mull of Kintyre in Scotland. A Board of Inquiry concluded that the most probable cause of the crash was an inappropriate rate of climb although no evidence existed that either pilot was negligent. Both were killed and could not answer for themselves and the Board of Inquiry verdict was overturned by two Royal Air Force reviewing officers who had decided that the pilots were grossly negligent.

So what has this got to do with "Computer says …"? Well, there were official doubts about the quality of the FADEC (Fully Automated Digital Engine Control) in the engines with engineers at the UK Ministry of Defence's own testing site at Boscombe Down in England now known to have described this system as "positively dangerous"[6]. There was more. Two years before the crash, a Royal Air Force Chinook Airworthiness Review Team was established because of safety and maintenance concerns of this aircraft. In spite of this and a fault code which was showing in the self-diagnosis unit of a FADEC system recovered from the wreck[7], the Royal Air Force steam-rollered its own rules that dead pilots could be found negligent only if there was "absolutely no doubt whatsoever".

The crash occurred in 1994. After a major campaign not dissimilar to the Post Office fiasco fought every inch of the way by both the Royal Air Force and the UK Ministry of Defence, the pilots were finally exonerated 17 years later in 2011.

## The potential impact of AI

In this Impact column, we have covered the evolution and implementation of real systems in a very wide variety of contexts. All of these systems were produced by traditional software development methods but we have entered a new era, that of "Artificial Intelligence"[8].

The particular aspect of AI we would like to address is its ability to cloak attempts to find out what a system is doing so effectively as to defy normal forensic analysis at worst or at best make it considerably more difficult than it already is. A further confounding factor is that 'the computer says' is now replaced by 'one server out of the millions of servers of a giant company that our

[5] Mr. Bates v. the Post Office; 4 episodes from 1st January 2024.

[6] https://www.bbc.com/news/uk-scotland-glasgow-west-14125241, accessed 30-May-2024.

[7] Fastidious journalism produced by https://www.computerweekly.com/news/2240089594/Chinook-crash-critical-internal-memo-on-software-flaws.

[8] https://www.ibm.com/topics/artificial-intelligence, accessed 30-May-2024.

lawyers do not dare to mess with and one which may not even be willing to tell us where the offending computer and its software is located'.

To our knowledge, there are no known occurrences as yet of a deployed AI system being involved in a widely publicized "Computer says …" incident but it is early days so we will have to imagine how this might develop knowing what we do about the training of such systems.  In essence an AI system is a thick mat of interconnected software nodes weighted perhaps in a transient way according to input training data and trained to produce certain outputs from its inputs.  It is then used to make decisions about inputs which were not part of its training data.  So, we can ask the question, "If an unusual and perhaps ridiculous result emerges from such a system, how easy is it to find out why?".   In general diagnosing software systems which have failed in order to produce some causal link between the failure (an end-user experience) and the fault or faults which caused it deep in the code, is computationally difficult and all too frequently impossible.  There is no general theory of such diagnosis and debugging code can be famously difficult.  In contrast, AI systems are generally somewhat simpler algorithmically but are likely to be exceedingly complex from a data point of view.  It's fair to say that there is no satisfactory theory of forensic understanding of failures in such systems available.  Search terms for "diagnosing failures in AI systems" or similar reveal at best references to the use of poor training data.  This however is not good enough.  If the lessons of the UK Post Office above are to be taken seriously, the default position for interactions with computer systems, whether incorporating AI or not, should be not be one of blind trust.  All engineering systems can be fallible.  All software engineering systems probably are.  It is up to us to understand those limitations carefully as we move forward with these extraordinary technologies.

## ***References***

Adams, E.N. (1984) "Optimizing Preventive Service of Software Products", IBM Journal of Research and Development, 28(1), pp 2-14.

Hatton, L. (2024) "Dependable Software Depends on Dependable Measurement", IEEE Computer, 57, pp 57-67.