Abstract for https://www.usenix.org/sites/default/files/healthtech15_cfp.pdf
August 10, 2015, Washington DC

Safety hazards in clinical calculators and apps

Harold Thimbleby / harold@thimbleby.net
College of Science
Swansea University
Swansea, Wales, SA2 8PP

Albert Wu / awu@jhsph.edu
Center for Health Services and Outcomes Research
Johns Hopkins Bloomberg School of Public Health
624 North Broadway
Baltimore, Maryland 21205, USA

Keywords: safety, apps, calculators

Abstract

When Thomas Nicely publicized a defect in the Intel Pentium, the media and public reaction was dramatic, despite Intel claiming the defect had little practical impact. Why then are more common reproducible and preventable defects in HIT not causing a similar backlash? Clinicians regularly perform calculations, a purpose for which computers were invented. Yet many calculators are flawed and may induce errors that clinicians will typically be blamed for.

Keying errors on calculators are not detected. A patient died after two nurses omitted to divide by 24 in a dose calculation. One optimal sequence of keystrokes for a popular calculator is [AC] [MRC] [MRC] 4 × 24 [M+] [AC] 5250 ÷ 45•57 ÷ [MRC] = but omitting the 24 simply gets the wrong answer with no error reported (it is also strange that a calculator needs so many complex keystrokes for routine calculations). This solution also relies on a nurse knowing the shortcut a/b/c = a/(b*c), which is needed on a calculator without brackets.

Our eye tracking experiments show users fixate on the keypad more than the display, yet many calculators if they display an error warning only show it fleetingly, so the user is very unlikely to notice it. If the user proceeds with the calculation, the final result will be incorrect but there will be no warning it is. Some calculators have delete keys, which do not work correctly. Some calculators keep logs of calculations, but these are logs of the calculation not what the user keys — so when there are bugs in the calculator, these are not recorded or detectable in the log. There are many other problems that are reproducible.

Perhaps worst there is an unnecessary variety between calculator designs, so the same sequences of keystrokes on one will give different results on another. This will induce unnecessary use error.

Medical apps also suffer from a wide range of problems. One allows a user to enter a time 5 mins in the future and then calculates a (fatal!) drug dose of 24 L over 5 minutes (144 L per hour). The software behaves as if the device's clock and the user's entered times might be up to five minutes out (a reasonable assumption), so times are rounded, but not consistently throughout a calculation.

Unfortunately, calculators are not evaluated for effectiveness (they are assumed to be good) and most apps are evaluated, if at all, for usability and clinical effectiveness in very small trials (e.g., 30 minutes × N=34) and human participant trials are far too short to detect the sorts of serious bugs highlighted here.

These are IT problems that are completely avoidable, yet are certainly causing unnecessary patient harm.

Even this very basic HIT is too immature for safe use in hospitals. The problems are soluble by formal methods and (more easily) stochastic simulation, which is a more accessible approach for programmers unable to use formal methods.