Errors + bugs needn't mean death

Healthy interaction design will save lives…

Human error and software bugs combine to lead to many adverse medical incidents, and even cause deaths. Both human error and software bugs are like microbes – they are hard to see, hard to understand, and therefore they are often ignored. People do not realise things could be a whole lot better.

Many patients used to die from infection in hospitals. Nobody understood bugs. Then in the 1840s Ignaz Semmelweis tried to bring in cleanliness to control infection in maternity hospitals, but his ideas were rejected. The establishment's resentment of Semmelweis ran so deep that he was tricked into a mental institution where he died from, ironically, an infected wound. The 'Semmelweis reflex' has now gone down in history as the rejection of knowledge when it contradicts entrenched views. People did not realise things could be better.

Semmelweis's ideas were rejected because of widespread ignorance about infection and misplaced confidence in status and traditional methods. Things had to wait until the research of Louis Pasteur, Robert Koch and Joseph Lister, whose work resulted in vastly improved survival rates for surgery and childbirth, as well as in vaccinations and our modern ideas of infection, immunity and hygiene.

Human error is very like a microbe. It isn't understood, and it kills people. It is estimated that preventable errors in hospitals in Western countries kill more people than are killed on the roads.

Moreover, like microbes, human error is invisible, and people like to pretend it does not affect them. Errors in hospitals are vastly underreported. We think good people

Our research shows numeric keypads (shown) are more error-prone than up/down number entry keys

cannot make errors, just like surgeons didn't think they carried infection.

Admitting making an error is a sign of weakness. People who do admit making errors are at best suspended or moved on, thus leaving behind a team who 'do not make errors' and have no experience of error management. In pharmacy it is illegal to dispense the wrong drug, so a pharmacist making a slip may lose their job. Pharmacy students therefore never get lecturers who have made (or admitted to making) errors, thus perpetuating the view that errors are rare and the exception.

IT systems and high technology are promoted as the solution to any number of modern ills. Well-known computer systems like Amazon, Facebook and Google are amazingly effective. But our dreams about IT solutions often ignore error. Systems are specified on the basis of what managers think is happening in hospitals, and then the computer systems cannot handle error, since managers so rarely know about it. In fact, error happens all the time and teams work together to manage the errors: people are very resilient in the face of error, but not when computer systems do not let them manage the errors that inevitably happen. Very quickly failed computer systems disappear from our consciousness and we never think again about their design bugs.

Our research is showing that errors are exacerbated by computer systems and by computerised medical devices, like drug infusion pumps.

In an incident in 2006, a cancer patient was given a fatal overdose of a chemotherapy drug. The patient had an infusion pump automatically deliver drugs, but it was incorrectly programmed by a nurse and gave the patient an overdose.

Thinking of the problem as 'human error' misses the point. The nurses who made 'the' mistake are only a part of a much larger system. Lots of things went wrong. The infusion pump was very complex and hard to use. Two nurses worked out the drug dose, but they made an arithmetic error. The pharmacy printed over-complex instructions for the drug, which did not help the nurses. A calculator was used to do the arithmetic, and calculators are notorious for being hard to use. In this incident, the nurses should have divided the dose by 24 hours a day (to get an hourly dose), but there was a slip and they did not, thus giving a dose 24 times too high. The calculator could have been designed to help detect the error, yet the calculator they used did not notice, as it was a general-purpose calculator that had no idea what calculation was being done. The calculator was a weak link, and it failed to help the nurses detect the error they had made.

Why was the hospital using such a complicated process? Why was the infusion pump so hard to use? Why did the infusion pump manufacturer not properly design the pump to reduce errors? Why were ordinary general-purposes calculators used, which have no concept of drug doses? Why were the pharmacy instructions to the nurses so complex?

The story is sad, but it illustrates many important points. Hardly anybody outside of our specialised research field is thinking about safer design of hospital processes and devices. The report into the incident recognised that the pump was badly designed, but it did not notice its software bugs, and it did not comment that its after-the-fact review of the device should have been done by the manufacturers in the first place, as they would have then made a much safer device. Indeed, redesign of infusion devices and calculators could make a huge difference. In fact, one could get rid of calculators in hospitals altogether, as modern infusion pumps could be reprogrammed to do the necessary calculations directly and with much better safety checks.

We have made a study of number entry error. Virtually no system, from medical devices through to office applications like spreadsheets, handles error sensibly. For example, if a user keys in 2.3.4 (which is an obvious error), virtually no system detects this error so that the user can correct it. Instead, something unexpected happens, like it is treated as 23.4 – a number that may or may not be what was intended. Worse, if the device keeps a log (it may not), the log will generally show the user keyed in 23.4 rather than the device failed to detect the slip. This could result in the user being blamed for the device's bad design. People, however well trained, will always eventually make a slip and key something incorrect. Eventually, an error like 2.3.4 will be keyed, and on almost all devices, the result will be quite unpredictable, except that the device will allow the user to continue, and get the wrong result – possibly ending up with an adverse incident.

This problem is ubiquitous; there is virtually no device that handles number entry errors satisfactorily, and the result is widespread undetected errors.

An iPhone calculator developed in our lab detects and blocks over 30 possible types of user error that conventional approaches miss

Medically, what are classified as 'out by 10' errors are dangerous, whereas smaller errors, say 'out by two', have less effect in treatment. Out by 10 errors are very common, and most often caused by decimal point errors or accidentally keying in an extra zero on the end of a number. Fortunately, as our research shows, proper design can halve the out by 10 error rate. That is, if the user intended to key in 2.34, blocking the automatic default that gets it treated as 23.4 stops the patient getting a drug dose that would be 10 times too high. It is accurate to call such bad design 'buggy', and its consequences are as damaging as medical bugs.

Just like an infected wound can appear misleadingly healthy until it goes nasty, bug-ridden devices look fine until things go wrong. A manufacturer may not notice that there are bugs in the design, and they will stay there as latent software bugs 'waiting' for a nurse to make a slip during use. This is a bit like an infection lying dormant in somebody until circumstances, such as stress, cause the latent infection to turn into an illness. As with medical infections, the solution with software and user interface 'infection' is primarily to avoid the problems arising in the first place, and, secondarily, when problems do arise, to diagnose them correctly. Our research shows that medical devices do have serious bugs and that our healthcare culture is effectively in denial.

Just as hand washing would have avoided deaths in Semmelweis's time, better interaction design today would save lives – but it is getting just as sceptical a response as hand washing did.

Our research is funded by a programme grant, 'Computer-Human Interaction in Medicine' (www.chi-med.ac.uk), from EPSRC, the UK Engineering and Physical Sciences Research Council. The project involves a multidisciplinary team in a large multi-university collaboration: Swansea, UCL, Queen Mary and City Universities. We have open jobs and PhD studentships and a Master's course, and we would particularly love to hear from manufacturers who want to improve the usability and safety of their products. We also welcome visiting researchers and industrial secondments.

Further Reading – available from www.harold.thimbleby.net

H Thimbleby, 'Is IT a dangerous prescription?' BCS Interfaces, 84, pp. 5-10, 2010

H Thimbleby and P Cairns, 'Reducing Number Entry Errors: Solving a Widespread, Serious Problem', Journal Royal Society Interface, 7(51), pp. 1429-1439, http://harold.thimbleby.net/interface, 2010

Professor Harold Thimbleby
Future Interaction Technology Lab

Department of Computer Science
Swansea University
Swansea SA2 8PP

Tel: +44 (0)1792 602299

www.harold.thimbleby.net
www.fitlab.eu
www.chi-med.ac.uk