ISSN 1754-3657 (Print)
ISSN 1754-3665 (Online)

CS4FN

Computer Science for Fun

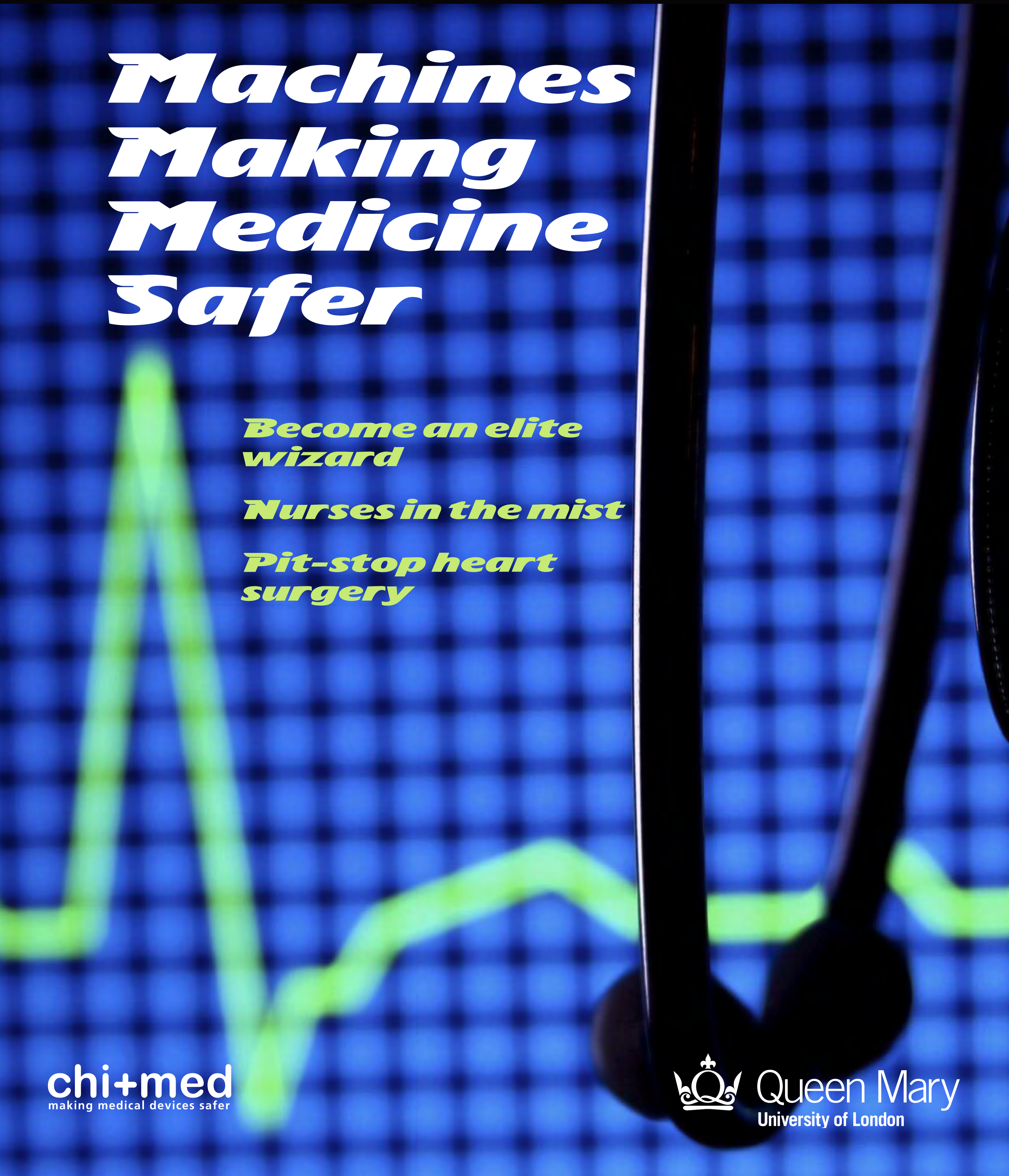
Issue 17

Machines Making Medicine Safer

Become an elite wizard

Nurses in the mist

Pit-stop heart surgery



chi+med
making medical devices safer

Queen Mary
University of London

Machines Making Medicine Safer

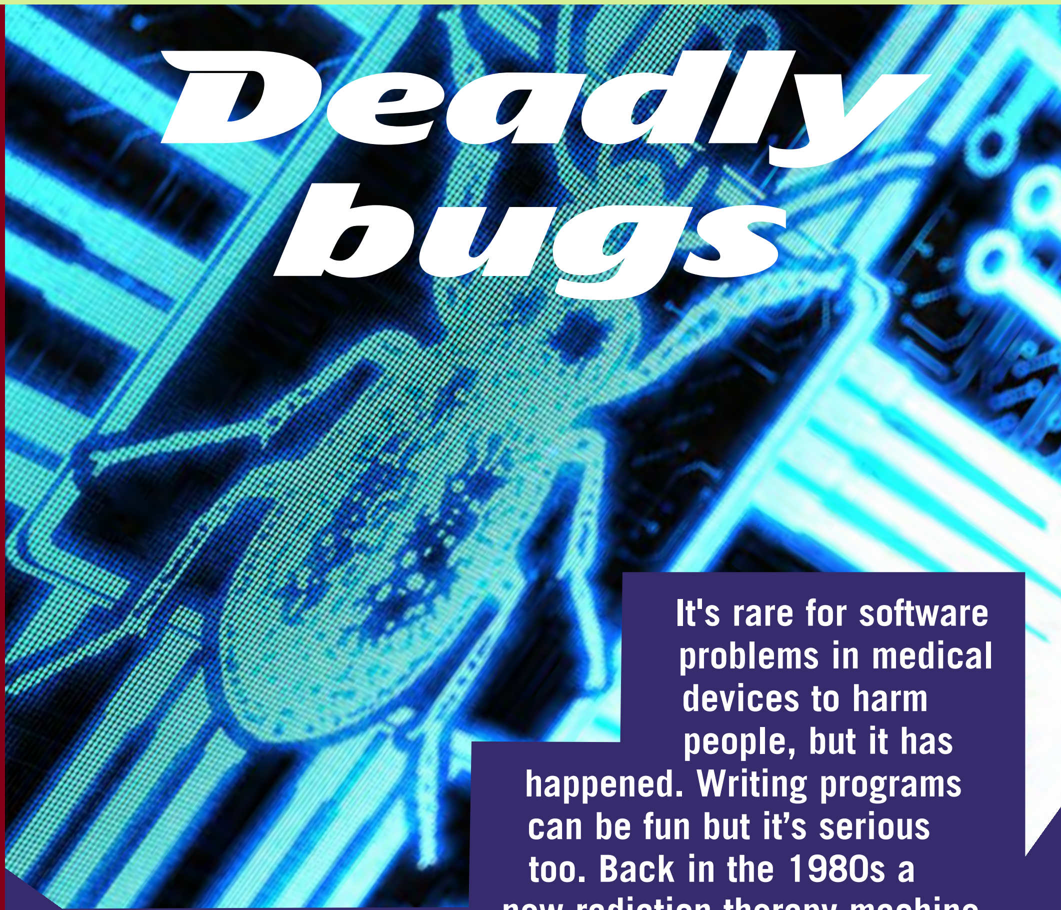
Computer scientists save lives. Our hospitals are full of computers in disguise, helping medics keep patients alive.

We explore the world of safe medical machines through a London-Swansea research project ‘CHI+MED’. Computer scientists are working with psychologists, social scientists and medical professionals to make medical devices even safer.

To build machines that not only save lives but are safe you need to think about the bigger picture, not just the gadgets. Programming them is important but it is only a small part of what matters. Only when you know how they are really used can you find ways to make them even safer.

We will also see how a Formula 1 team have contributed to safer operations, how taking a leaf out of a gorilla watcher’s book has helped one device manufacturer get a market edge, and why programmers really are wizards!

Deadly bugs



It's rare for software problems in medical devices to harm people, but it has happened. Writing programs can be fun but it’s serious too. Back in the 1980s a new radiation therapy machine called Therac-25 was released. It was an improvement on an earlier machine. They were designed to zap cancer with a beam of radiation. Radiation is dangerous, which makes that risky business. But, with the right dose, it can cure cancer!

Therapy machines include safety systems to ensure patients aren’t accidentally given an overdose. The earlier Therac-20 had a mechanical system that made it physically impossible for a high-powered beam to be directed at the patient. For the new model the physical system was replaced by software - ironically because it was thought to be safer and more advanced. It could produce a low energy beam of electrons, but also a high energy beam of x-rays by firing high-powered electrons at a metal plate.

Unfortunately, because of a bug - a programming mistake - it was sometimes possible for the complex software to get the timing off and do things in the wrong order - this is known as a ‘race condition’. They are really hard bugs to find. It meant that the high powered beam could be fired without the protective metal guard being in place and the patient could be given an overdose. It turned out that the operating system software used by the machines had been created by a programmer without proper training. It should never have been used in a device where safety mattered.

At least some good has come from it. It showed that safety critical software has to be designed to be more than just safe. It has to be designed so you can check that it really is safe. Therac-25 also highlighted how important it is that all programmers writing software used in systems that could harm people are trained in safety-critical code development, understand safety and use appropriate methods to write and check code.

Programming Wizards

Elite programmers are often called 'wizards'. It turns out that isn't far from the truth!

A magic trick is made of two parts: a secret method and a way of presenting it. Unless both work, the magic trick will fall flat. Software is similar. It also consists of two things. The first is an algorithm: the instructions that when followed by the computer lead to it doing whatever it's there to do. The second is the human-computer interface: the code that makes it easy to use. It turns out algorithms are just like magic secret methods and interfaces are like the presentation of a trick.

Many magic tricks are 'self-working'. That means they don't involve sleight of hand or hidden mechanisms. As long as you follow the steps the magical effect occurs and your audience is left surprised. But that is what computer scientists mean by an 'algorithm'. Programs are just algorithms written down in a language a computer can follow. Self-working tricks are algorithms for doing magic. In fact the algorithms of some tricks are based on ones actually used in computing applications. For example, there are powerful tricks based on the algorithm used by early computers to search for data.

The whole audience make the same mistake at the same time

Most tricks have an underlying algorithm, but also include steps that rely on some kind of sleight of hand or misdirection. The audience look in the wrong place at the wrong time, or are made to believe something that isn't true. They don't see everything, even if everything is there to be seen. The magician engineers the magic 'system' so that the whole audience make the same mistake at the same time.

It is of course similarly possible to engineer a computer system so that its users make mistakes too. Unlike a magician programmers don't do it on purpose. Some human errors are not about negligence or stupidity but are about the limitations of our brains and the way we see the world. That is exactly what a magician relies on. Software developers have to do the opposite. They have to design the system so that no one makes mistakes in using them. If they don't take human limitations into account their programs will mislead.

Misdirection, is one way a magician misleads. It relies on the fact that we can only focus our attention on one small area at a time. If it is drawn to one point then we will miss other things. Instead of drawing a person's attention away from things that matter, with software, their attention should be drawn to those critical things. For example, if a nurse mistypes a dose into a machine delivering a drug, then before starting the infusion, you want the nurse's eye drawn back to the screen, not away from it, using the magician's tricks, so they check the dose is as expected.

Both programmers and magicians rely on misdirection

Whether designing an algorithm or an interface, the principles are the same as those behind creating a great magic trick. When programmers are writing new programs they are using the same computational thinking skills as magicians inventing new tricks. Learn to program and you could become a real wizard!

Try a magic trick that relies on misdirection for yourself and see how easily people make mistakes. Go to Magazine+ on the cs4fn website **www.cs4fn.org** and look for the Four Aces trick.

The red so of DOO

A red sock in with your white clothes wash – guess what happened next? What can you do to prevent it from happening again? Why should a computer scientist care? Jo Brodie investigates. It turns out that red socks have something to teach us about medical gadgets.

How can we stop red socks from ever turning our clothes pink again? We need a strategy. Here are some possibilities.

- Don't wear red socks.
- Take a 'how to wash your clothes' course.
- Never make mistakes.
- Get used to pink clothes.

Let's look at them in turn - will they work?

Don't wear red socks: That might help but it's not much use if you like red socks or if you need them to match your outfit. And how would it help when you wear purple, blue or green socks? Perhaps your clothes will just turn green instead.

Take a 'how to wash your clothes' course: Training might help: you'd certainly learn that a red sock and white clothes shouldn't be mixed, you probably did know that anyway, though. It won't stop you making a similar mistake again.

Never make misteaks: Just never leave a red sock in your white wash. If only! Unfortunately everyone makes mistakes - that's why we have erasers on pencils and a delete key on computers - this idea just won't work.

Get used to pink clothes: Maybe, but it's not ideal. It might not be so great turning up to school in a pink shirt.

What if the problem's more serious?

We can probably live with pink clothes, but what happens if a similar mistake is made at a hospital? Not socks, but medicines. We know everyone makes mistakes so how do we stop those mistakes from harming patients? Special machines are used in hospitals to pump medicine directly into a patient's arm, for example, and a nurse needs to tell it how much medicine to give - if the dose is wrong the patient won't get better, and might even get worse.

What have we learned from our red sock strategies? We can't stop giving patients medicine and we don't want to get used to mistakes so our first and fourth strategies won't work. We can give nurses more training but everyone makes mistakes even when trained, so the third suggestion isn't good enough either and it doesn't stop someone else making the same mistake.

We need to stop thinking of mistakes as a problem that people make and instead as a problem that systems thinking can solve. That way we can find solutions that work for everyone. One possibility is to check whether changes to the device might make mistakes less likely in the first place.

Errors? Or arrows?

Most medical machines are controlled with a panel with numbered keys (a number keypad) like on mobile phones, or up and down arrows (an arrow keypad) like you sometimes get on alarm clocks. CHI+MED researchers have been asking questions like: which way is best for entering numbers quickly, but also which is best for entering numbers accurately? They've been running experiments where people use different keypads, are timed and their mistakes are recorded. The researchers also track where people are looking while they use the keypads. Another approach has been to create mathematical descriptions of the different keypads and then mathematically explore how bad different errors might be.

It turns out that if you can see the numbers on a keypad in front of you it's very easy to type them in quickly, though not always correctly! You need to check the display to see if you have actually put in the right ones. Worse, mistakes that are made are often massive - ten times too much or more. The arrow keypads are a little slower to use but because people are already looking at the display (to see what numbers are appearing) they can help nurses be more accurate, not only are fewer mistakes made but those that are made tend to be smaller.

Image credit - Flickr user DomFurniss

ck

M

Smart machines help users

A medical device that actively helps users avoid mistakes helps everyone using it (and the patients it's being used on!). Changing the interface to reduce errors isn't the only solution though. Modern machines have 'intelligent drug libraries' that contain information about the medicines and what sort of doses are likely and safe. Someone might still mistakenly tell the machine to give too high a dose but now it can catch the error and ask the nurse to double-check. That's like having a washing machine that can spot bright socks in a white wash and that refuses to switch on till it has been removed.

Building machines with a better ability to catch errors (remember, we all make mistakes) and helping users to recover from them easily is much more reliable than trying to get rid of all possible errors by training people. It's not about avoiding red socks, or errors, but about putting better systems in place to make sure that we find them before we press that big 'Start' button.

Pit-stop he

The Formula 1 car screams to a stop in the pit-lane. Seven seconds later, it has roared away again, back into the race. In those few seconds it has been refuelled and all four wheels changed. Formula 1 pit-stops are the ultimate in high-tech team work. Now the Ferrari pit stop team have helped improve the hospital care of children after open-heart surgery!

Open-heart surgery is obviously a complicated business. It involves a big team of people working with a lot of technology to do a complicated operation. Both during and after the operation the patient is kept alive by computer: lots of computers in fact. A ventilator is breathing for them, other computers are pumping drugs through their veins and yet more are monitoring them so the doctors know how their body is coping.

Pass it on

One of the critical times in open-heart surgery is actually after it is all over. The patient has to be moved from the operating theatre to the intensive care unit where a 'handover' happens. All the machines they were connected to have to be removed, moved with them or swapped for those in the intensive care unit. Not only that, a lot of information has to be passed from the operating team to the care team. The team taking over need to know the important details of what happened and especially any problems, if they are to give the best care possible.

A research team from the University of Oxford and Great Ormond Street Hospital in London wondered if hospital teams could learn anything from the way other critical teams work. This is an important part of computational thinking – the way computer scientists solve problems. Rather than starting from scratch, find a similar problem that has already been solved and adapt its solution for the new situation.

Just as the pit-stop team are under intense time pressure, the operating theatre team are under pressure to be back in the operating theatre for the next operation as soon as possible. In a handover from surgery there is lots of scope for small mistakes to be made that slow things down or cause problems that need to be fixed. In situations like this, it's not just the technology that matters but the way everyone works together around it. The system as a whole needs to be well designed and pit stop teams are clearly in the lead.

Smooth moves

To find out more, the research team watched the Ferrari F1 team practice pit-stops as well as talking to the race

art surgery

director about how they worked. They then talked to operating theatre and intensive care unit teams to see how the ideas might work in a hospital handover. They came up with lots of changes to the way the hospital did the handover.

For example, in a pit-stop there is one person coordinating everything – the person with the 'lollipop' sign that reminds the driver to keep their brakes on. In the hospital handover there was no person with that job. In the new version the anaesthetist was given the overall job for coordinating the team. Once the handover was completed that responsibility was formally passed to the intensive care unit doctor. In Formula 1 each person has only one or two clear tasks to do. In the hospital people's roles were less obvious. So each person was given a clear responsibility: the nurses were made responsible for issues with draining fluids from the patient, anaesthetist for ventilation issues, and so on. In Formula 1 checklists are used to avoid people missing steps. Nothing like that was used in the handover so a checklist was created, to be used by the team taking on the patient.

These and other changes led to what the researchers hoped would be a much improved way of doing handovers. But was it better?

Calm efficiency saves the day

To find out they studied 50 handovers – roughly half before the change was made and half after. That way they had a direct way of seeing the difference. They used a checklist of common problems noting both mistakes made and steps that proved unusually difficult. They also noted how well the teams worked together: whether they were calm and supported each other, planned what they did, whether equipment was available when needed, and so on.

They found that the changes led to clearly better handovers. Fewer errors were made both with the technology and in passing on information. Better still, while the best performance still happened when the teams worked well, the changes meant that teamwork problems became less critical.

Pit-stops and open-heart surgery may be a world apart, with one being about getting every last millisecond of speed and the other about giving as good care as possible. But if you want to improve how well technology and people work together, you need to think about more than just the gadgets. It is worth looking for solutions anywhere: children can be helped to recover from heart surgery even by the high-octane glitz of Formula 1.

Image credit: Flickr user rafa_uoc

Much ado about nothing

The nurse types in a dose of 100.1 mg of a powerful drug and presses start. It duly injects 1001 mg into the patient without telling the nurse that it didn't do what it was told. You wouldn't want to be that patient!

Designing a medical device is difficult. It's not creating the physical machine that causes problems so much as writing the software that controls everything that that machine does. The software is complex and it has to be right. But what do we mean by "right"? The most obvious thing is that when a nurse sets it to do something, that is exactly what it does.

Getting it right is subtler than that though. It must also be easy to use and not mislead the nurse: the human-computer interface has to be right too. It is the software that allows you to interact with a gadget – what buttons you press to get things done and what feedback you are given. There are some basic principles to follow when designing interfaces. One is that the person using it should always be clearly told what it is doing.

Manufacturers need ways to check their devices meet these principles: to know that they got it right.

It's not just the manufacturers, though. Regulators have the job of checking that machines that might harm people are 'right' before they allow them to be sold. That's really difficult given the software could be millions of lines long. Worse they only have a short time to give an answer.

Million to one chances are guaranteed to happen.

Problems may only happen once in a million times a device is used. They are virtually impossible to find by having someone try possibilities to see what happens, the traditional way software is checked. Of course, if a million devices are bought, then a million to one chance will happen to someone, somewhere almost immediately!

Paolo Masci at Queen Mary University of London, has come up with a way to help and in doing so found a curious problem. He's been working with the US regulator for medical devices – the FDA – and developed a way to use maths to find problems. It involves creating a mathematical description of what critical parts of the interface program do. Properties, like the user always knowing what is going on, can then be checked using maths. Paolo tried it out on the code for entering numbers of a real medical device and found some subtle problems. He showed that if you typed in certain numbers, the machine actually treated them as a number ten times bigger. Type in a dose of 100.1 and the machine really did set the dose to be 1001. It ignored the decimal point because on such a large dose it assumed small fractions are irrelevant. However another part of the code allows you to continue typing digits. Worse still the device ignores that decimal point silently. It doesn't make any attempt to help a nurse notice the change. A busy nurse would need to be extremely vigilant to see the tiny decimal point was missing given the lack of warning.

A useful thing about Paolo's approach is that it gives you the button presses that lead to the problem. With that you can check other devices very quickly. He found that medical devices from three other manufacturers had exactly the same problem. Different teams had all programmed in the same problem. None had thought that if their code ignored a decimal point, it ought to warn the nurse about it rather than create a number ten times bigger. It turns out that different programmers are likely to think the same way and so make the same mistakes (see 'Double or Nothing', right).

Now the problem is known, nurses can be warned to be extra careful and the manufacturers can update the software. Better still they and the regulators now have an easy way to check their programmers haven't made the same mistake in future devices. In future, whether vigilant or not, a nurse won't be able to get it wrong.

Mathematical tools can be hard to use. Paolo with Enrico D'Urso from Queen Mary and Patrick Oladimeji from Swansea have developed a front-end that allows interfaces to be easily created and tested behind the scenes.

Double or nothing?

If you spent billions of dollars on a gadget you'd probably like it to last more than a minute before it blows up. That's what happened to a European Space Agency rocket. How do you make sure the worst doesn't happen to you? How do you make machines reliable?

A powerful way to improve reliability is to use redundancy: double things up. A plane with four engines can keep flying if one fails. Worried about a flat tyre? You carry a spare in the boot. These situations are about making physical parts reliable. Most machines are a combination of hardware and software though. What about software redundancy?

You can have spare copies of software too. Rather than a single version of a program you can have several copies running on different machines. If one program goes wrong another can take over. It would be nice if it was that simple, but software is different to hardware. Two identical programs will fail in the same way at the same time: they are both following the same instructions so if one goes wrong the other will too. That was vividly shown by the maiden flight of the Ariane 5 rocket. Less than 40 seconds from launch things went wrong. The problem was to do with a big number that needed 64 bits of storage space to hold it. The program's instructions moved it to a storage place with only 16 bits. With not enough space, the number was mangled to fit. That led to calculations by its guidance system going wrong. The rocket veered off course and exploded. The program was duplicated, but both versions were the same so both agreed on the same wrong answers. Seven billion dollars went up in smoke.

Can you get round this? One solution is to get different teams to write programs to do the same thing. The separate teams may make mistakes but surely they won't all get the same thing wrong! Run them on different machines and let them vote on what to do. Then as long as more than half agree on the right answer the system as a whole will do the right thing. That's the theory anyway. Unfortunately in practice it doesn't always work. Nancy Leveson, an expert in software safety from MIT, ran an experiment where different programmers were given programs to write. She found they wrote code that gave the same wrong answers. Even if it had used independently written redundant code it's still possible Ariane 5 would have exploded.

Redundancy is a big help but it can't guarantee software works correctly. When designing systems to be highly reliable you have to assume things will still go wrong. You must still have ways to check for problems and to deal with them so that a mistake (whether by human or machine) won't turn into a disaster.

Nurses in the Mist

What do you do when your boss tells you "go and invent a new product"? Lock yourself away and stare out the window? Go for a walk, waiting for inspiration? System engineers Pat Baird and Katie Hansbro did some anthropology.

Dian Fossey is perhaps the most famous anthropologist. She spent over a decade living in the jungle with gorillas so that she could understand them in a way no one had done before. She started to see what it was really like to be a gorilla, showing that their fierce King Kong image was wrong and that they are actually gentle giants: social animals with individual personalities and strong family ties. Her book and film, 'Gorillas in the Mist', tells the story.

Pat and Katie work for Baxter Healthcare. They are responsible for developing medical devices like the infusion pumps hospitals use to pump drugs into people to keep them alive or reduce their pain. Hospitals don't buy medical devices like we buy phones, of course. They aren't bought just because they have lots of sexy new features. Hospitals buy new medical devices if they solve real problems. They want solutions that save lives, or save money, and if possible both! To invent something new that sells you ideally need to solve problems your competitors aren't even aware of. Challenged to come up with something new, Pat and Katie wondered if, given the equivalent was so productive for Dian Fossey, perhaps immersing themselves in hospitals with nurses would give the advantage their company was after. Their idea was that understanding what it was really like to be a nurse would make a big difference to their ability to design medical devices. That helped with the real problems nurses had rather than those that the sales people said were problems. After all the sales people only talk to the managers, and the managers don't work on the wards. They were right.

Taking notes

They took a team on a 3-month hospital tour, talking to people, watching them do their jobs and keeping notes of everything. They noted things like the layout of rooms and how big they were, recorded the temperature, how noisy it was, how many flashing lights and so on. They spent a lot of time in the critical care wards where infusion pumps were used the most but they also went to lots of other wards and found the pumps being used in other ways. They didn't just talk to nurses either. Patients are moved around to have scans or change wards, so they followed them, talking to the porters doing the pushing. They observed the rooms where the devices were cleaned and stored. They looked for places where people were doing ad hoc things like sticking post it note reminders on machines. That might be an opportunity for them to help. They looked at the machines around the pumps. That told them about opportunities for making the devices fit into the bigger tasks the nurses were using them as part of.

The hot Texan summer was a problem

So did Katie and Pat come up with a new product as their boss wanted? Yes. They developed a whole new service that is bringing in the money, but they did much more too. They showed that anthropology brings lots of advantages for medical device companies. One part of Pat's job, for example, is to troubleshoot when his customers are having problems. He found after the study that, because he understood so much more about how pumps were used, he could diagnose problems more easily. That saved time and money for everyone. For example, touch screen pumps were being damaged. It was because when they were stored together on a shelf their clips were scratching the ones behind. They had also seen patients sitting outside in the ambulance bays with their pumps for long periods smoking. Not their problem, apart from it was Texas and the temperature outside was higher than the safe operating limit of the electronics. Hospitals don’t get that hot so no one imagined there might be a problem. Now they knew.

Porters shouldn’t be missed

Pat and Katie also showed that to design a really good product you had to design for people you might not even think about, never mind talk to. By watching the porters they saw there was a problem when a patient was on lots of drugs each with its own pump. The porter pushing the bed also had to pull along a gaggle of pumps. How do you do that? Drag them behind by the tubes? Maybe the manufacturers can design in a way to make it easy. No one had ever bothered talking to the porters before. After all they are the low paid people, doing the grunt jobs, expected to be invisible. Except they are important and their problems matter to patient safety.

The advantages didn't stop there, either. Because of all that measuring, the company had the raw data to create models of lots of different ward environments that all the team could use when designing. It meant they could explore in a virtual environment how well introducing new technology might fix problems (or even see what problems it would cause).

All in all anthropology was a big success. It turns out observing the detail matters. It gives a commercial advantage, and all that mundane knowledge of what really goes on allowed the designers to redesign their pumps to fix potential problems. That makes the machines more reliable, and saves money on repairs. It's better for everyone.

Talking to porters, observing cupboards, watching ambulance bays: sometimes it's the mundane things that make the difference. To be a great systems designer you have to deeply understand all the people and situations you are designing for, not just the power users and the normal situations. If you want to innovate, like Pat and Katie, take a leaf out of Dian Fossey's book. Try anthropology.

A wee story

'Father Christmas needs a wee' is a classic book that teaches kids to count. It's all about something that secretly fascinates: how close do other people get to wetting themselves? It's just a story but it inspired Dom Furniss of UCL to tell us some festive, but true, stories about his friends. Read on to learn something useful about avoiding nasty accidents.

Going, going, gone

Rob was heading home from London after the office Christmas party. Emma, a work friend, offered him a lift home. He sat in the middle of the back seat with a girl from work on either side. He still had a can of drink to finish, but maybe bringing it wasn't such a good idea. Not too long into the journey he began to feel the need to wee! It wasn't that far so surely he'd be able to hold on, though.

Unfortunately, the feeling grew stronger. Beads of sweat formed on his brow as he fought to hold the pee back. The others in the car were chatting and laughing. He didn't want to be a hassle and ask them to pull over, and besides they were on a motorway, so it'd be difficult anyway. Weighing everything up Rob thought the best course of action would be just to let a little bit out, which would relieve the pain and buy him time to have a proper wee when he got home. You can probably guess the rest... he sat there and watched his plan go wrong as a dark wet patch grew and grew. Once started nothing could stop it.

The relief paled into insignificance as the gravity of the situation sank in. He was in his colleague's car, sat between two people from work, and without a whimper he had wet himself. However, this will surprise you: after such a stupid mistake Rob now performed an amazing act of genius - he faked his elbow being pushed by one of the girls and dropped his can of drink into his lap. She apologised profusely, he told her not to worry. He smiled smugly as he finished the journey not as the perpetrator of a pee crime but as the victim of an unfortunate spillage.

The knowledge

What can this tell us about the science behind human error? Well this is an example of a 'knowledge-based error'. They happen when a person has some wrong or missing knowledge about a situation. In this case Rob should have known that you can't just let a little bit out. How could he not know? Surely, everyone knows that if you're dying for a pee, once you pop you ain't stopping until it's finished! The trouble is something that is obvious to one person can be unknown to another. If Rob had the correct knowledge he would have chosen a different course of action. Unfortunately he didn't.

Preventing this kind of problem is about making sure people are well-trained and have all the knowledge they need to do the job. This is also where learning from mistakes is vital. If you make a knowledge-based error and understand why it happened, you should never make it again. You now have the correct knowledge. Ideally, others should learn from your mistake too, though, if you don't hide it like Rob did.

In work situations, once we know of a problem, we can train people so they have the knowledge to choose the right courses of action. If rather than looking to blame or shame when someone makes a mistake we look for lessons to learn, we can prevent the problem happening to anyone in the future.

Risky baths

This is especially important with medical technology. Take the tragic case of the patient who died because of taking a hot bath. They had a patch inserted under their skin that delivered the drug they needed in a low-hassle way. What they didn't know was that a hot bath increases how much drug is delivered by the patch. If we raise awareness of this danger it will hopefully prevent others patients doing the same. A tragic accident like this can also lead to a review of wider issues than just the particular thing that patient didn't know. For example, reviewing how patients are told about this sort of safety information more generally might lead to other kinds of accidents being avoided.

That patient knew nothing about the risks of taking a hot bath. But suppose they did know. It's still entirely plausible that they could take a hot bath having just forgotten that they shouldn't. Part of the advantage of these patches is that they can be implanted then virtually forgotten about. What is supposed to be a blessing turns into a curse. You mustn't forget about the patch if you are thinking about a bath. This is a completely different kind of error to a knowledge-based error. It's called a ‘slip error’ and it needs a different kind of solution to prevent it. It needs innovative design not just the spreading of knowledge, but that's another wee story.

For a wee story about slips, go to www.cs4fn.org

Why not contribute to research and log your own everyday errors. Go to www.errordiary.org

Erorrdiary streams funny, frustrating and fatal errors. Whether exotic or mundane and everyday, it aims to raise awareness about how errors are all around us, no matter how clever we pretend to be.

Oh no! Not again...

What a mess. There's flour all over the kitchen floor. A fortnight ago I opened the cupboard to get sugar for my hot chocolate. As I pulled out the sugar, it knocked against the bag of flour which was too close to the edge... Luckily the bag didn't burst and I cleared it up quickly before anyone found out. Now it's two weeks later and exactly the same thing just happened to my brother. This time the bag did burst and it went everywhere. Now he's in big trouble for being so clumsy!

Was it his fault? Should he have been more careful? He didn't choose to put the sugar in a high cupboard with the flour.

Maybe it was my fault? I didn't choose to put the sugar there either. But I didn't tell anyone about the first time it happened. I didn't move the sugar to a lower cupboard so it was easier to reach either. So maybe it was my fault after all? I knew it was a problem, and I didn't do anything about it.

Now think about your local hospital.

James is a nurse, working in intensive care. Penny is really ill and is being given insulin by a machine that pumps it directly into her vein. The insulin is causing a side effect though - a drop in blood potassium level - and that is life threatening. They don't have time to set up a second pump, so the doctor decides to stop the insulin for a while and to give a dose of potassium through a second tube controlled by the same pump. James sets up the bag of potassium and carefully programs the pump to deliver it, then turns his attention to his next task. A few minutes later, he glances at the pump again and realises that he forgot to release the clamp on the tube from the bag of potassium. Penny is still receiving insulin, not the potassium she urgently needs. He quickly releases the clamp, and the potassium starts to flow. An hour later, Penny's blood potassium levels are pretty much back to normal: she's still ill, but out of danger. Phew! Good job he noticed in time and no-one else knows about the mistake!

Two weeks later, James' colleague, Julia, is on duty. She makes a similar mistake treating a different patient, Peter. Except that she doesn't notice her mistake until the bag of insulin has emptied. Because it took so long to spot, Peter needs emergency treatment. It's touch-and-go for a while, but luckily he recovers.

Julia reports the incident through the hospital's incident reporting system, so at least it can be prevented from happening again. She is wracked with guilt for making the mistake, but also hopes fervently that she won't be blamed and so punished for what happened.

Report it!

Alexis Lewis, at Swansea University, has been exploring how best to design incident reporting forms as part of her PhD. Lots of different forms are used in hospitals across the UK and she examined more than 20. Many had features that would make it harder than necessary for nurses and doctors to report incidents. Some failed to ask about important facts and many didn't encourage feedback. It wasn't clear how much detail or even what should be reported. She is using the results to design a new reporting form that avoids the problems and that can be built into a system that encourages the reporting of incidents so that hospital staff learn from the incidents that do happen.

Whodunnit?

To stop different people making the same mistakes over and over again you need to know why they happen. Incident investigators try to work that out. They act like detectives digging up and reviewing evidence. The difference is that they are less interested in whodunnit than in what led to it happening. One way they do that is called root cause analysis. You try and track back beyond the immediate reason something went wrong to the ultimate causes. For example, if a doctor gave someone the wrong drug, it might be that they misread the name or that the two drugs were stored together so the wrong one easily picked up. Once they know why an incident happened, investigators make recommendations about how these real causes can be prevented. Perhaps certain drugs should not be stored together, for example.

The methods used help the investigator focus on the things that matter. Huayi Huang, a PhD student at Queen Mary, has developed a new way to to do this. Rather than focus on causes, his method concentrates on the way information travels around - from person to person and between people and machines. At each point the idea is to look at what safety measures are in place to make sure that information doesn't go wrong. Rather than focus on whodunnit, focus on why nothing stopped it happening!

Don't miss the near misses

Why did it happen? There are a whole bunch of problems that are nothing to do with Julia. Why wasn't it standard practice to always have a second pump set up for critically ill patients in case such emergency treatment is needed? Why can't the pump detect which bag the fluid is being pumped from? Why isn't it really obvious whether the clamp is open or closed? Why can't the pump detect it. If the first incident - a 'near miss' - had been reported perhaps some of these problems might have been spotted and fixed. How many other times has it happened but not reported?

What can we learn from this? One thing is that there are lots of ways of setting up and using systems, and some may well make them safer. Another is that reporting "near misses" is really important. They are a valuable source of learning that can alert other people to mistakes they might make and lead to a search for ways of making the system safer - but only if people tell others about them. Reporting near-misses can help prevent the same thing happening again.

The above was just a story, but it's based on an account of a real incident... one that has been reported so it might just save lives in the future.

Microwave Racing

When you go shopping for a new gadget like an MP3 player are you mostly wowed by its sleek looks, or drool over its long list of extra functionality? Do you then not use those extra functions because you don't know how? Rather than just drooling, why not go to the races to help find a device you will actually use.

On your marks, get set... microwave

Take an everyday gadget like a microwave. They have been around a while, so manufacturers have had a long time to improve their designs and make them easy to use. You wouldn't expect there to be problems would you? There are lots of ways a gadget can be harder to use than necessary - more button presses maybe, lots of menus to get lost in, more special key sequences to forget, easy opportunities to make mistakes, no obvious feedback to tell you what it's doing... Just trying to do simple things with each alternative is one way to check out how easy they are to use. How simple is it to cook some peas with your microwave? Could it be simpler? Dom Furniss, a researcher at UCL decided to video some microwave racing as a fun way to find out.

Better design helps avoid mistakes

Everyday devices still cause people problems even when they are trying to do really simple things. What's clear from Microwave racing is that some really are easier to use than others. Does it matter? Perhaps not if it's just an odd minute wasted here or there cooking dinner or if actually, despite your drooling in the shop, you don't really care that you never use any of those 'advanced' features.

Would it matter to you more if the device were in a hospital keeping a patient alive and where a mistake could harm them? There are lots of gadgets like this: infusion pumps for example. They are the machines you are hooked up to in a hospital via tubes. They pump life-saving drugs, nutrient rich solutions or extra fluids to keep you hydrated directly into your body. If the nurse makes a mistake setting the rate or volume then it could make you worse rather than better. Surely then you want the device to help the nurse get it right.

While the consequences are completely different, the core thing you do in setting an infusion pump is actually very similar to setting a microwave - "set a number for the volume of drug and another for the rate to infuse it and hit start" versus "set a number for the power and another for the cooking time, then hit start". The same types of design solutions (both good and bad) crop up in both cases. Nurses have to set such gadgets day in day out. In an intensive care unit, they will be using several at a time with each patient. Do you really want to waste lots of minutes of such a nurse's time, day in, day out? Do you want a nurse to easily be able to make mistakes?

User feedback

What the microwave racing video shows is that the designers of gadgets can make them trivially simple to use. They can also make them very hard to use if they focus more on the looks and functions of the thing than ease of use. Manufacturers of devices are only likely to take ease of use seriously if the people doing the buying make it clear that we care. Mostly we give the impression that we want features so that is what we get. Microwave racing may not be the best way to do it, but next time you are out looking for a new gadget check how easy it is to use before you buy... especially if the gadget is an infusion pump and you happen to be the person placing orders for a hospital!

Go to www.cs4fn.org to see Microwave racing in action and find out how the pros really evaluate gadgets.

Screaming headline kills!!!

Most people in hospital get great treatment but if something does go wrong the victims often want something good to come of it. They want to understand why it happened and be sure it won't happen to anyone else. Medical mistakes can make a big news story though with screaming headlines vilifying those 'responsible'. It may sell papers but it could also make things worse.

If press and politicians are pressurising hospitals to show they have done something, they may just sack the person who made the mistake. They may then not improve things meaning the same thing could happen again if it was an accident waiting to happen. Worse if we're too quick to blame and punish someone, other people will be reluctant to report their mistakes, and without that sharing we can't learn from them. One of the reasons flying is so safe is that pilots always report 'near misses' knowing they will be praised for doing so, rather than getting into trouble. It's far better to learn from mistakes where nothing really bad happens than wait for a tragedy.

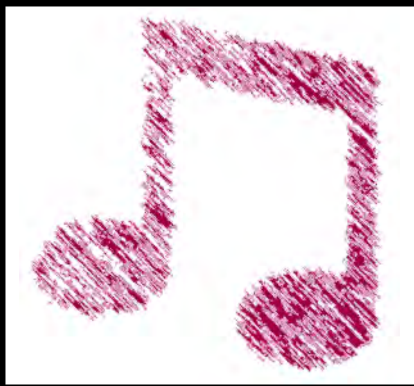
Share mistakes to learn from them

Chrystie Myketiak from Queen Mary is exploring whether the way a medical technology story is reported makes a difference to how we think about it, and ultimately what happens. She analysed news stories about three similar incidents in the UK, America and Canada. She wanted to see what the papers said, but also how they said it. The press often sensationalise stories but Chrystie found that this didn't always happen. Some news stories did imply that the person who'd made the mistake was the problem (it's rarely that simple!) but others were more careful to highlight that they were busy people working under stressful conditions and that the mistakes only happened because there were other problems. Regulations in Canada mean the media can't report on specific details of a story while it is being investigated. Chrystie found that, in the incidents she looked at, that led to much more reasoned reporting. In that kind of environment hospitals are more likely to improve rather than just blame staff. How the hospital handled a case also affected what was written - being open and honest about a problem is better than ignoring requests for comment and pretending there isn't a problem.

As we've seen everyone makes mistakes (if you don't believe that, the next time you're at a magic show, make sure none of the tricks fool you!). Often mistakes happen because the system wasn't able to prevent them. Rather than blame, retrain or sack someone its far better to improve the system. That way something good will come of tragedies.

Pumpin Blood

Tunetrace is a popular app that turns drawings and photographs into a musical light show. Now this award winning idea has gone into showbiz. With Warner Records and pop band NoNoNo, a new Tunetrace app has been released that allows you to resequence the band's track, Pumpin Blood, based on the photograph or drawing you give it. The app extracts the pattern of line crossings in the picture. That pattern is treated as a sort of computer program. Depending on secret rules it uses it to jump the track in beat steps to create new musical masterpieces. Why not have a play, download it for free from www.qappsonline.com/apps/pumpin-blood/



Playing Bridge, but not as we know it

Clifton, Forth and Brooklyn are all famous suspension bridges where, through a feat of engineering greatness, the roadway hangs from cables slung from sturdy towers. The Human Harp project created by Di Mainstone, Artist in Residence at Queen Mary, involves attaching digital sensors to bridge cables attached by lines to the performer's clothing. As the bridge vibrates to traffic and people, and the performer moves, the angle and length of the lines are measured and different sounds produced. In effect human and bridge become one augmented instrument, making music mutually. FInd out more at www.humanharp.org

Think of a number

If you add 1 to 1219 you get 1220, except sometimes you get 1210. What's going on? It all depends on what the numbers mean and what they're doing. It happens to you all the time and you probably don't bat an eyelid.

A typical padlock will only open when the four-digit combination is dialed in so that the digits line up correctly. Each disc of digits is separate from the other three and turning one doesn't affect any of the others. That makes sense - when you choose a combination you are picking 4 completely separate digits.

Clocks go forward

With a typical digital clock on the other hand the digits can be changed separately but they also act as if they are linked. As the time approaches 7am the hour digit won't change from '6' until the two minute digits have reached 59 (the clock now reads 6:59). At precisely 7 o'clock the two digits on the right will say 00 but the hour digit will also move on from 6 to 7 so that the time reads 7:00. If you were setting an alarm for 7am you might be glad that all three digits act together, and not just the minute ones at the end - otherwise it would be easy to accidentally set your alarm for 6 o'clock instead of 7!


credit: Flickr user IndieCade

With the padlock when you add 1 to a combination 1219 you get the combination 1210. With the clock, adding 1 to the time 12:19 you get 12:20. You need to know what your numbers mean and so how they work if you are to change them correctly.

Can you hack it

There are lots of ways to enter numbers into gadgets, choosing the right one for the job at hand can make a big difference. Perhaps there are ways no one has thought of before! At a hack day run by Gerrit Niezen of Swansea University, the CHI+MED research team played with different ways of entering numbers, and developed a bunch of new ideas to try out. For example, Sarah Wiseman from University College London turned three Sifteo Game Cubes, an interactive toy, into devices for entering numbers. Smaller than a matchbox the cubes have a clickable screen that displays games, or in this case numbers, and they're sensitive to movement and position, a bit like Wii game controllers. Sarah programmed two of the cubes to display one number each and the third cube to act as an up/down button to change the numbers. She also told the number cubes to behave as separate digits (like the padlock) if the controller was placed above or below them, or as linked numbers (like the alarm clock) if the controller was placed to the left or right of the cubes. Knowing whether the numbers in your machine act separately or together can make the difference between an extra hour in bed but it also helps patients to get the right dose of medicine.

Go to the cs4fn website to watch Sarah's video about numbers on YouTube.

Turning the world upside down

We've all got used to using mobiles: whether tablets, phones or music players. Mobile computers are changing the way we do just about everything. Now hospitals are in on the act. But sometimes turning the world upside down can be a problem, especially if you are left-handed.

Many hospitals now use mobile devices to treat patients. Machines that are small and light enough to move around with a patient mean that those patients don't have to be stuck in bed. Some can even take them home and have their treatment there: a much nicer experience. But this comes with some risks. Because mobile devices can move they can also end up upside down, and numbers can go wrong when they're the wrong way up.

If you've got a device strapped to your arm then someone else looking at it will see the numbers correctly but if you look at it you'll read them upside-down. If you look at them in the mirror you'll see the numbers back to front. What numbers look like matters - you have to be able to tell what they are and which way up they are. The numbers 6 and 9, and 2 and 5, can easily be confused depending on which way up they are and how they're written.

Hand-held devices... but which hand?

People with diabetes use a pen-shaped device to give them insulin. These have a number display built into them. You rotate a dial at the end of the pen to change the dose. Most people are right-handed so hold the pen in their left hand and rotate the dial with their right. One of the early insulin pens had a number display with a design flaw that meant you couldn't actually tell which way up the numbers were. It would show the correct dose if you were right handed but showed the wrong dose to left-handed people, who were holding the pen 'upside down'. Unless you know which way is meant to be 'up' you can't tell if you're going to give a dose of 6 or 9 units, or even 12 instead of 21? Fortunately the problem was noticed and because the lesson was learned, insulin pens are now much safer.

CHI+MED's Harold Thimbleby from Swansea University has been worrying about this kind of thing, and how to make sure numbers on medical devices are less likely to cause problems. A common way for gadgets to display numbers is to use a '7 segment display'. The number 8 uses all 7 segments, made up of vertical and horizontal bars. By either showing or hiding other segments you can make up all the numbers from 0 to 9. They can be part of the problem though because many digits on a 7 segment display are identical to others when flipped.

Harold has pointed out that while 7 segment displays are simple and great for some things they're terrible for machines where safety is critical. It might not be a matter of life or death if you don't know if your clock is saying 12:51 or 15:21 (turn the magazine upside down to see) but getting numbers the wrong way round in medicine is never a good idea. Technology that turns the world upside down sometimes needs to be used with care.

In an episode of Jonathan Creek ('The mother redcap') the hero realises that a witness told the police the wrong time. She thought she saw 5:10 on the clock when disturbed in the middle of the night, but a glass of water in front of the hour digit reversed the number from 5 to 2, so the real time was 2:10.

Back (page) in the running

Health and wellbeing are near the top of everyone's to do list, even if some of us don't actually turn it into action. Researchers around the world are exploring how technology can help us be more sporty and healthy. After all, prevention is better than cure. So let's run through a few gadgets aiming to keep you healthy.

Steps will have to be taken

Accelerometers, tiny devices that measure force and movement, are common in smartphones and wearable computing. The Fitbit and Nike+ devices, for example, can detect how many steps you take and how many stairs you climb. You can set your daily targets or just measure your general activity and share the results with friends. You can even share them with the world in general, so keeping fit becomes a global game. For those less physical, accelerometers are also used in special 'sleep phase' alarm clocks. They monitor body movement and wake you up when you're in light sleep mode. That makes waking up easier and more pleasant.

And the winner is – stairs... eventually

Walk virtually anywhere

Omni aim to take game playing a step further. They are developing a platform that you stand on and can walk on. As you walk, wearing special shoes, the platform floor detects your footfall. That movement is translated into movement of your character on screen. The system will replace a normal games controller. Instead of pressing buttons, it's your actual body walking and running that does the controlling. As computer games increase in their realism this could become the new (if tiring) way to get around that virtual town.

And the winner is – World of Warcraft taxi companies

Bike snatcher catcher

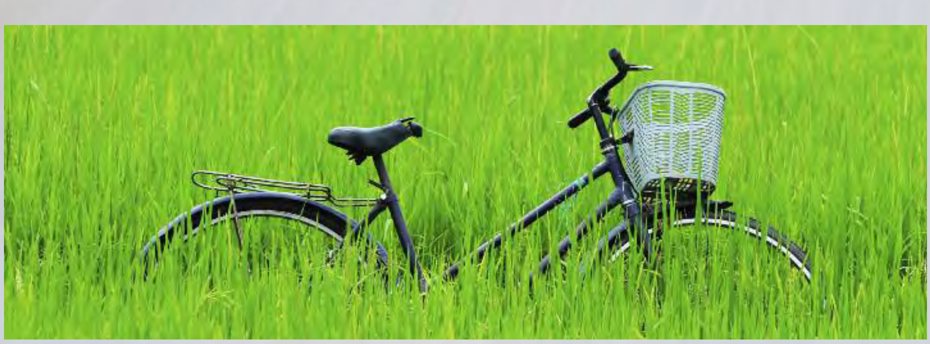
Cycling is a popular way to get fit and help the environment. An American start-up company have created Bike+, a gadget for your bike. It logs the distances you have peddled and the routes taken, so you can use this data to get fitter. It also has an extra twist; one of the things about having a smart bike is that it can be a target for thieves. Bike+ also acts as an alarm, so if the bike is moved when it shouldn't be it sets off an alarm and sends you a text. You can track where the thieves have taken your beloved bike, and hopefully recover it safely.

And the winner is – the long arm of the law and a good pair of legs

Gear of good cheer

A good run can relieve stress, but researchers at Microsoft have come up with a new way to monitor how stressed you are: the smart bra. It measures the heartbeat, skin moisture and activity of the person wearing it, and in tests was able to predict stress levels pretty accurately. Microsoft are now looking to develop something more unisex, in the form of a bracelet. Someday soon you clothes or fashion accessories might be telling you to take a run.

And the winner is – pushy clothes, take a hike



Editors: Paul Curzon, Jo Brodie, Chrystie Myketiak and Peter McOwan, 2014
EPSRC supported this issue through CHI+MED which consists of UCL, Swansea, Queen Mary University of London and City Universities. Google support cs4fn.

Queen Mary University of London

chi+med making medical devices safer | Department for Education | SUPPORTED BY MAYOR OF LONDON | EPSRC Engineering and Physical Sciences Research Council | COMPUTING AT SCHOOL EDUCATE · ENGAGE · ENCOURAGE