Taking a hard line on software

It shouldn't come as a surprise to you that aeroplanes are heavily reliant on computer software. So are many other items that we literally trust with our lives, like medical facilities and the braking systems in cars. These machines all rely on software to perform their life-or-death functions.

Now consider the software on your computer, think of the number of times that software has failed you. Would you trust your computer software with your life? If the answer is no, then why trust the software in anything, especially the software in that essential computer we use more than anything else: the mobile phone?

How do you verify the software in mobile phones? Well as usual, ANU has the solution. Dr Michael Norrish is one of the researchers that successfully verified mobile phone operating systems. Having earned his Ph.D. from Cambridge University, he carefully explains the process of verifying mobile phone operating systems:

Imagine you wanted to test that software didn't crash at an inappropriate moment. In most disciplines, you would run the software as many times as possible and track the number of errors. You would then reduce errors by adjusting and trialling again. But unfortunately, with aeroplanes and medical equipment trial and error isn't an option. You have to test all the possibilities.

Instead, Dr Norrish uses mathematical theorem proving to verify the software.

Phones have two operating systems that allow other software to access the hardware. One of these is for calls, and the other is for everything else. With one mobile phone operating system consisting of 10,000 lines of code, and the software on top of that consisting of 1 million lines of code, testing all the possibilities would take a lifetime, so logic is the only way to verify the software.

Dr Norrish and his colleagues have to prioritise which software to verify, so how do you do that? Well, think back to your laptop, when one piece of software crashes the other software starts crashing too. So if the operating system for phone calls crashes, the billing, network and satellite software might crash also, making it an instant priority.

So is helping industry solve a commercial dilemma worthy ANU research? Is it a contribution to human knowledge, for the benefit of the nation? Absolutely!

Being somewhat modest, Dr Norrish claims that the project was more of an accomplishment than the creation of new knowledge. But by succeeding, the techniques they used to reduce mobile phone software errors were proven correct. Now proven, other academics will build on their techniques and apply them in other areas of software verification, thus, the new knowledge is embedded in the process.

When asked about the contribution of his research to "the stock of human knowledge" he framed his answer in a considered way: that the project took year and thousands of man hours to complete. 3 of his team members received their Ph.D. for the project. Norrish remarks almost as an afterthought that the project developed these 3 colleagues to a state of readiness. They are now capable of pursuing new knowledge and they will succeed.

An interesting concept, certainly one you wouldn't expect from anyone other than an academic of mathematic theorem proving: by preparing other young minds for the difficulties of research one can make a greater contribution to human knowledge. It takes humility. It doesn't carry prestige. But in the end, that "stock of human knowledge" is greater.

This unorthodox means of growing the stock of knowledge by growing the number of knowledge creators as well as researching is reserved for those with a true passion for the pursuit of new knowledge.

Cameron Bestwick (u4859411)