

Better than the best

DR NEVILLE GRECH writes about his research career in automating software

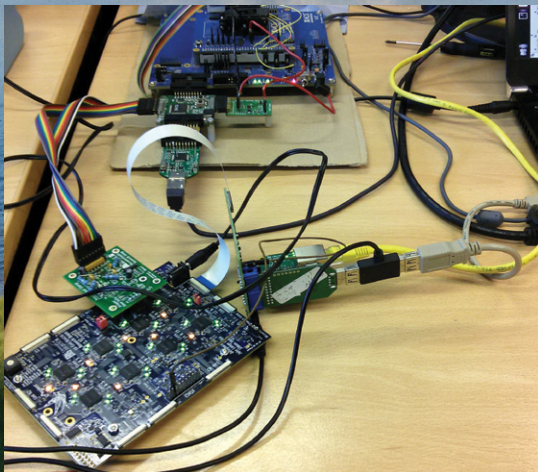
TODAY WE RELY on computers. Computers are present inside car engines, mobile phones, and even medical implants. Programming languages are the way humans can dictate how these machines behave. Programming a machine is difficult, involving a lot of trial and error until the machine starts behaving as the human programmer wants. Programmers spend their time testing programs; however, despite all their efforts, most programs that are used in production systems are still error laden.

Languages such as JavaScript, called dynamically-typed languages, have gained a strong foothold amongst programmers. These languages enable writing software quickly. The JavaScript language is ubiquitous on the web, and dynamically-typed languages are very common in the software industry, including Malta's. Unfortunately, these programs are also littered with errors (called type errors) that can break an entire system. Programmers would benefit by detecting type errors in their code at an early stage, without imposing

unnecessary restrictions on their programs. Due to their dynamic nature, no such tools exist. For my Ph.D. research I chose to fill this gap by developing a mechanism called "pre-emptive type checking", that can automatically find type errors early on. It can be applied to languages such as JavaScript or Python. In fact, I have developed a proof of concept implementation for the Python language, one of the most commonly used programming languages. Various research institutions are working on different ways to make dynamically-typed languages more reliable, with Microsoft Research (MSR) spearheading this kind of research. Researcher Gavin Bierman, working for MSR (now Oracle), has worked on a recent programming language called TypeScript, together with a number of others in use today. Bierman was my external examiner for my Ph.D., and discussing these issues with him during my viva was a truly enriching and intellectually stimulating experience. Despite the large number of people working on the problem, I have managed to tackle the problem from a

different angle. My work focused on creating a mechanism to find type errors as quickly as possible. The earlier a mistake is detected, the less damage is caused and the earlier one can fix it.

Following my Ph.D. in Southampton, I moved to the University of Bristol to pursue post-doctoral research on compilers, which are programs that turn human readable programs into machine-readable code. Computer energy consumption is a major industry challenge, and the software is a large drain on battery life. I collaborated with semiconductor companies and other top research institutions to develop software tools that estimate energy consumption when running different programs. I also developed techniques to automate the process of writing energy efficient software. Software that is more dependable, maintainable, and more efficient than the best systems currently available. ●



A glimpse of the future? An embedded computer system, with 256 efficient processor cores running at 500Mhz. My research at Bristol with XMOS is helping with the automation of writing energy efficient software for this, and similar, platforms.

