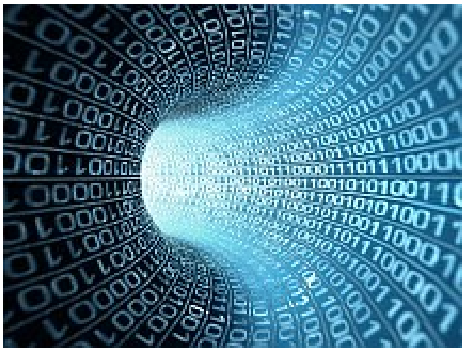
PHYS.ORG

# Goodbye to faulty software?

July 15 2008



Will it ever be possible to buy software guaranteed to be free from bugs? A team of European researchers think so. Their work on the mathematical foundations of programming could one day revolutionise the software industry.

We have become used to the idea that software will not work properly. While we would take a faulty car back to the dealer and demand they put it right, we are remarkably tolerant of software that goes wrong.

The software we buy usually comes with no guarantee and disclaimers are notoriously all encompassing. We no longer expect everything to work correctly ‘out of the box’. More to the point, neither does the manufacturer. Indeed, software houses seem to rely on their customers to find faults, which they can then ‘patch’ in a so-called ‘upgrade’ of the product.

“The software industry is still very immature compared to other branches of engineering,” says Dr Bengt Nordström, a computer scientist at Chalmers University in Göteborg. “We want to see programming as an engineering discipline but it’s not there yet. It’s not based on good theory and we don’t have good design methods to make sure that at each step we produce something that’s correct.”

Nordström believes that the whole approach to software design needs to be rethought. The usual approach is to validate a program via a lengthy testing process. Instead, he would like to see a design philosophy that guarantees from first principles that a program will do what it says on the box.

The key lies in an esoteric reformulation of mathematics called ‘type theory’ based on the notion of computation. In this approach, the specification for a computational task is stated as a mathematical theorem. The program that performs the computation is equivalent to the proof of the theorem. By proving the theorem the program is guaranteed to be correct.

## Open source

It is not that simple, of course, but so promising is type theory that since 1989 the EU has been funding a string of projects to develop it under the Future and Emerging Technologies programme.

Nordström was coordinator of one of the projects, TYPES, which fosters co-operation on the topic among researchers at 15 European universities and research institutes, along with those at 19 associated academic and industrial organisations.

The TYPES partners are also releasing open source software packages that anyone can download, use and modify. These packages include several ‘proof editors’ that, in type theory, are the key to guaranteeing the correctness of programs.

Can such an abstract research area really lead to reliable, bug-free software?

“European research in this field is the strongest in the world,” Nordström points out. “Many computer programs are going wrong, they don’t work properly, and in the long run this research will help. This is a very slow process, it takes many years to get ideas from the universities into industry but I think it’s slowly taking place.”

The open source principle, says Nordström, is fundamental to what they are trying to achieve.

“It’s important that anyone can evaluate the code and check if it is correct, so it’s inherent in this project that what we are doing should be open so that it can be discussed by everybody.”

Results from type theory are already finding their way into other projects. The EU-funded Mobius project is developing methods, known as ‘proof-carrying code’, for downloaded programs to be certified as bug-free.

Meanwhile, a France-based company is using ideas from type theory to design secure embedded computer systems such as those used for smart cards. Further research is also under way in Japan.

## Theory, in practice

Researchers have also demonstrated the power of type theory by proving the classic ‘four colour’ theorem with one of the proof editors used in TYPES. Type theory is also finding application in the analysis of human language.

Nordström does not see type theory as being necessary for all programs, but there is a clear need for guarantees in critical systems in banking, for example. But type theory could also be important in the transport, defence and healthcare sectors, where mistakes can cost lives.

TYPES received funding from the EU's Sixth Framework Programme for research as a 'coordination action', which describes projects that aim to oil the wheels of co-operation rather than directly develop a new technology. TYPES interweaves both basic and applied research.

"That's one thing I find very, very interesting compared to other sciences," Nordström notes. "We are maybe 150 people working in this project and it's a mixture of very practical persons and very theoretical persons and there is a lot of exchange between them. I think that's very rare compared to other sciences."

He hopes that the work done under TYPES will ultimately allow programming to mature into a genuine engineering discipline with the same high standards and quality assurance now expected elsewhere in the engineering profession.

"A lot of effort is now spent on testing software," he says. "Very often programs are written quite quickly and then they are tested and changed and tested again, and so on. It's very unsystematic. This is not how we build bridges and highways.

That style of working is going to change so that we spend more effort on actually writing programs than testing them."

Provided by ICT Results

Citation: Goodbye to faulty software? (2008, July 15) retrieved 26 April 2024 from https://phys.org/news/2008-07-goodbye-faulty-software.html