



A Cork researcher has developed a program that can check the speed of other programs, writes Dick Ahlstrom

Timing is everything in love and war, they say, but a researcher in Cork has taken timing to new levels. He devised a way to predict how long it takes for a computer program to run, something of crucial importance if the program is flying a passenger aircraft.

"The main issue is to improve software timing," says University College Cork computer science lecturer, Dr Michel Schellekens. "The industry wants to know when certain processes in the computer end. You need to be absolutely sure the activities end at a given time."

Schellekens is director of Ceol, UCC's centre for efficiency-oriented languages. He and his team have devised a novel computer algorithm that allows him to calculate how fast a computer program can get a job done.

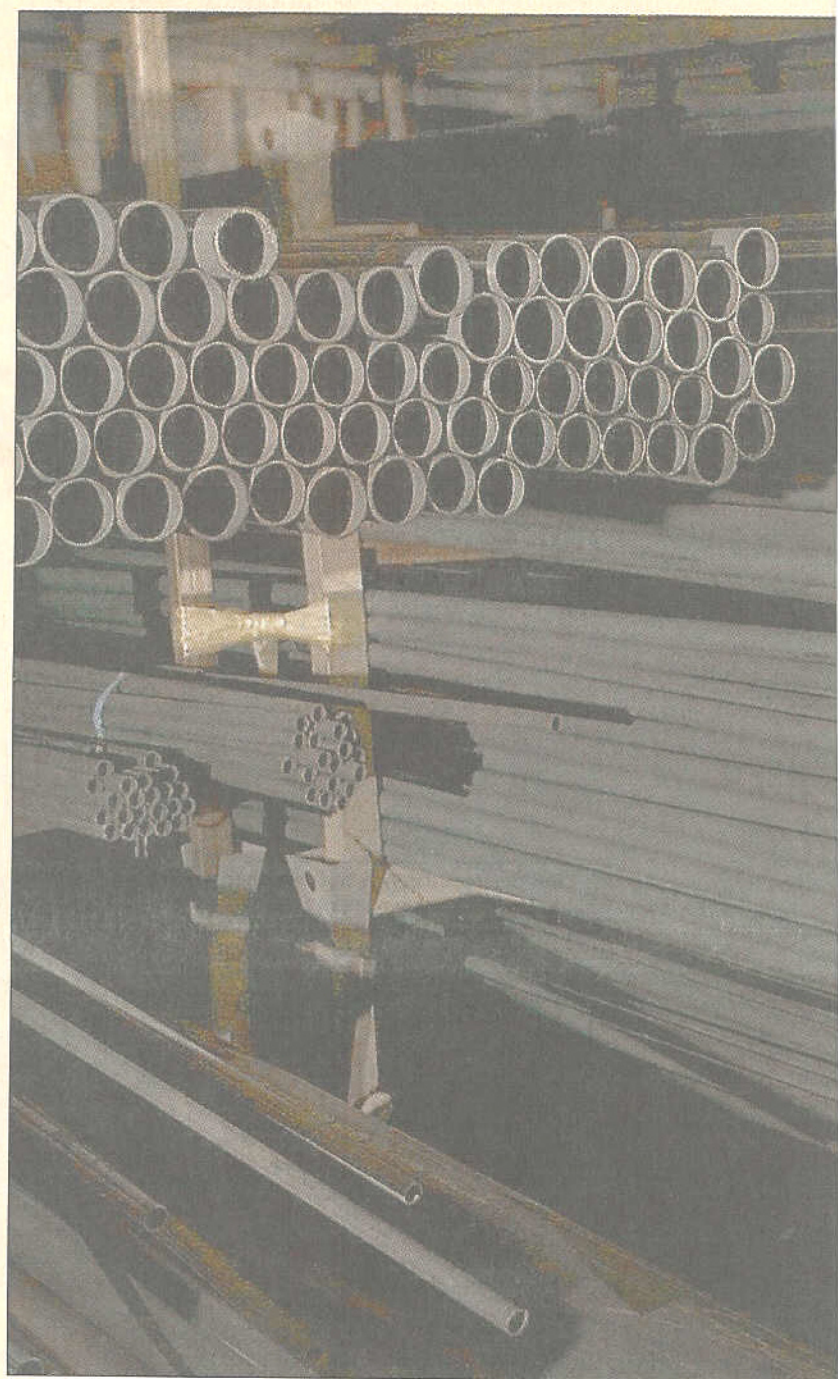
There is no room for approximations for some computer programmes, he explains. He mentions aircraft autopilot software but other time critical examples include the programme that controls anti-lock braking systems on the family car.

In these cases, the software has only fractions of a second to make a decision and initiate an action. Now Schellekens's algorithm can tell software designers how long the information processing will take.

Before now programmers only had approximations for the longest time required to run a programme to the end of an action. The new algorithm gives a more realistic average time no matter how complex the computer computation.

Software basically consists of a series of operations, where each operation takes a piece of information, transforms it and passes it on to the next operation for further manipulation, he says. In a sense the information flows through the computer like water flows through a pipe.

A computer can produce a list of the names of people on a given flight. If there are three people on the flight, the computer can order the names in six possible ways. The possible combinations grow rapidly as the number of passengers grows and by the time you reach 11 passengers, producing all possible name lists has already reached the processing limits for most personal



Clocking a

computers, Schellekens says.

For larger data sets there are too many possibilities to get good averages of processing time required, even if you run test data, he says. "You can't run through all of them. You need a mathematical tool to predict it."

Current computer timing systems are an average based on the "factorial" number, the number of items in the set. The 11 passengers are 11 factorial, he says. "If you go to 20 factorial you won't be able to compute that any more. That is the problem people face in calculating average time of a software run," he says.

If there were 65 passengers in a typi-

cal mid-sized airplane, that would produce more unique name lists than there are atoms in the universe.

Trying to produce a useable average of how long processing this information would take is a challenge with these large data sets. Rather than averaging the time taken for each permutation, Schellekens and his team developed a timing equation that can anticipate the average time needed to run a software programme. "This algorithm can calculate in any circumstance. It is comparable to Boolean algebra."

He calls his timing programme Acett, average-case execution time tool and



ll programs

Dr Michel Schellekens and his team developed a timing equation that can anticipate the average time needed to run a software program.

Photograph: Tomas Tyner

late last month he had a successful test run to prove that it works. Acett didn't need to watch the program running, it could derive running time directly from the programme's own code in a "modular way".

"Modularity basically means that there is great control over the predictability of the system in that the behaviour of the entire system can be derived from the behaviour of its basic components. This is at the root of engineering. To predict a system, one analyses its parts and derives the behaviour of the whole. Achieving it for software timing was a great research challenge which has been

overcome at Ceol."

Acett would be of interest to real-time programming language developers, he believes. It could be used for any company interested in guaranteeing that its software meets deadlines, such as the aviation and automotive industries, chemical plants, and robots.

"We are going to experiment with that and see if we can improve it," he says.

He hopes to do this both by refining Acett but also by developing new types of microchips that are hard-wired to run Acett. "We are aiming to develop a whole new processor based on that idea."