Embedded software made simpler yet more powerful

May 22 2006

The current decade will probably be known as the dawn of pervasive computing, when PCs were dethroned by technology to embed computers in almost everything. The hardware already exists to add features such as artificial intelligence and wireless connectivity to clothing or cars. Thanks to researchers, software is catching up fast.

"Hardware development has reached a stage where it is possible to have a fully-fledged computer with processor, memory and operating system on a board the size of a sliver of chewing gum," explains Germбn Puebla, a researcher at Madrid Technical University. "But until now software that can be programmed easily, and uses the limited hardware and power resources of pervasive computing devices as efficiently as possible has been lacking."

Puebla coordinated the ASAP project, which set out to solve the problem of creating and adapting software to run efficiently on pervasive computing systems, where computers are integrated in everyday objects and environments.

The result is a groundbreaking open source programming, analysis and optimisation toolkit for pervasive computing systems using Constraint Logic Programming (CLP) languages that has been validated in a series of case studies.

Until ASAP, the use of high-level CLP languages, which simplify programming and make software more portable across different platforms, had not been considered a feasible solution for pervasive systems because the convenience they provide to programmers comes at a cost: generally less efficient and more resource-hungry code.

Therefore, researchers have traditionally used low-level languages such as C, which tend to be more efficient but also more complicated to code, limits the versatility and complexity of the software, and generally forces programmers to manually rewrite the program for different platforms.

Because pervasive computing involves multiple different distributed platforms communicating among themselves the software needs to be interoperable, but, because of the limited processing and power resources of pervasive devices, most of which are battery operated, the software must also be as efficient as possible.

ASAP's toolkit offers a solution by using the high-level declarative language Ciao in a way that is optimised to reduce resource consumption.

"Software created with the toolkit is comparable in terms of resource demands to code written in C if it is designed to do the same thing. But Ciao programs can also do much more complex tasks, and with our toolkit it is feasible for them to run on pervasive systems," Puebla explains. "Ciao is also much easier to use – programmers don't have to reinvent the wheel every time they need to create or adapt a program."

Self-tuning and resource-aware analysis and specialisation algorithms allow the toolkit, dubbed CiaoPP, to produce specialised programs that are automatically optimised to meet particular processing and resource constraints. The CLP analysis and transformation tools can also act as a meta-language between a broad range of high and low-level languages to optimise and verify programs for pervasive computing.

Because of the automatic nature of the tools and the limited need for manual programming, the risk of errors being introduced into the code is also reduced.

"This is particularly important if we are looking at a future in which tiny computers are everywhere," Puebla notes.

In one of the project's case studies, pervasive application kernels written in Ciao were run on the gumstix single board computer – a chewing gum slice-sized computer – as part of a wearable computer system being developed by the University of Bristol, an ASAP project partner. They have already used the ASAP toolkit to develop software for a hearing device that can emulate the spatial and directional effects of sound, simulating, for virtual reality gaming, for example, the way we experience noises emanating from different sources in real life.

In industry, pervasive systems that monitor hazardous materials or the protective clothing of workers would improve safety and security. In healthcare, ubiquitous devices could keep check on patients' health remotely. And in the home, tiny computers embedded in everyday appliances could turn on the heating, dim the lights or even let you know when you are running low on milk.

"The uses for pervasive systems are almost infinite, and the market is potentially huge," Puebla notes.

Source: IST Results