Embedded systems get smarter, tougher

March 11 2008

RobuCab trundles towards tougher, smarter emdedded systems. Photo: Embounded

A European research team has achieved the twin, and apparently contradictory goals, of making embedded systems both smarter and tougher.

The RobuCab, an autonomous vehicle about the size of a golf cart, trundles at 10kph along a quiet French street. Alarmingly, it looks like it is driving itself. Surprisingly, that is more or less true.

The RobuCab is following the line of the kerb. One embedded system trains a camera on the path edge, another tracks the angle and direction of the kerb, while others control the gearing and acceleration. Combined, they enable the RobuCab to drive along the road.

It is an astonishing demonstration of just how sophisticated embedded

systems, and the software that controls them, can become. But there are some serious problems to surmount before this level of sophistication becomes common.

Embedded systems are all around us. They can range from very simple sensors that tell your boiler when to turn on the central heating in your home, to very powerful computers that help control flight.

“They are everywhere,” explains Kevin Hammond, coordinator of the Embounded project, a team developing sophisticated new software for the RobuCab. “Half the world’s annual spend on computers goes on embedded systems. And often, it is items we would not even think of as a computer, like a digital watch. But, like a digital watch, all embedded systems have software and some degree of processing hardware.”

They run ABS (‘anti-lock breaking systems’) in cars, avionics and high-tech toasters. They are in RFID (‘radio frequency identification’) chips, mobile phones and microwave ovens. Serious people are already talking seriously about ‘painting’ embedded systems onto walls just like, well, paint. Or of house bricks with microchips inside.

Calling 99.9999

But while embedded systems are tiny, industry and society makes huge demands on them. “Some of them, like avionics, must be essentially unbreakable, with six nines of uptime.” That means they must operate 99.9999 percent of the time. By contrast, desktop computers need only work 75 percent of the time.

“An embedded system’s memory might only run to 10 or 20 bytes of information, but these tiny systems must be more reliable than normal desktop computers,” Hammond states.

And that is only the first challenge. The tasks they are designed to do are becoming much more sophisticated, like the RobuCab, and that makes programming them extremely difficult.

Thus far, embedded systems were programmed using very simple instructions but, while these are powerful, hundreds of simple instructions are required to drive the more complex tasks of emerging systems. More instructions mean an exponential growth in the risk of error.

“Specialist engineers currently spend an enormous amount of time testing these systems, but, even then, there is no guarantee that all possible problems have been checked,” warns Hammond.

Paradoxical goals

As they become more sophisticated, they can do more complex tasks but the risk of failure grows. So Embounded began with two apparently paradoxical goals: establish precise controls to enhance safety and create a more sophisticated programming language at a higher level of abstraction. One that tells the system what goal to achieve, but does not tell it precisely how to do it.

Hammond explains that the team sought to improve precision and performance, but also wants to reduce programming control. “It is a non-trivial problem,” he jokes.

Embounded tackled this by first developing a new, more sophisticated programming language for embedded systems, called Hume. Next, it developed a programming methodology that increases system precision and performance using certificates to limit, or “Embound”, resource usage.

Then, they developed “costing-by-construction”, a technique to sandbox the functional modules within a computer program. This means they are kept apart, making it easier to guarantee the required resources for each functional module.

Finally, they developed a suite of tools to analyse prototype-embedded systems. This can guarantee that a given system design will work as planned.

It is an enormous number of outputs for a comparatively small project and the team’s work has inspired the enthusiasm of colleagues in research and industry. “Finally someone has combined the critical features needed for successful development of embedded systems: [Hume has] exactly the features I have been looking for … I have actually designed a language myself to accomplish some of the goals, but I can scrap that now since you seem to get it all right (unlike me),” wrote one Swedish researcher not associated with the project.

“We have had many more people contact us spontaneously, so clearly engineers and scientists are looking for a new way of developing embedded systems.

“Funding agencies, too, are very enthusiastic, and the consortium received further funds to work on software for an autonomous vehicle for the UK government. This was really a piece of blue sky research, we were not a commercialisation project. But we have developed a strong prototype and worked out where the shoe pinches,” says Hammond.

The project gained from a very high level of co-operation and synergy between the partners. LASMEA, in Clermont-Ferrand, used the RobuCab to test the system, while, AbsInt GmBH in Saarbrucken, Germany, produced high-quality execution time information. The Ludwig Maximilian University of Munich worked on resource

certification analysis, while Heriot-Watt University in the UK provided compilers and other tools.

The University of St. Andrews developed fundamental models and analyses, as well as overseeing the project. "The fit was very good, and we've developed strong links over the course of the project," says Hammond.

The Embounded team has submitted proposals for a follow-on EU-funded project. And the work on Hume and its associated methodologies and tools will carry on regardless, edging forward, like RobuCab, towards tougher and smarter embedded systems.

Source: ICT Results