hardware jocks who, after designing the device itself, decided to sling a few lines of code. The Wind River pair preferred to apply the recyclable approach to software writing they had honed at Berkeley Lab. It was an avant-garde methodology that they brought to customers like Coppola, for whom they developed a system that allowed film editing to begin in production via computer, and Lake Systems, a firm that was inventing a specialized video editor that allowed National Football League teams to review plays.

They also were able to see the potential in two new developments. The powerful Unix operating system, popularized in the early '80s, broke operations down into components and thus was the perfect development system for modular-designed applications. And the Ethernet hardware connection was going to be faster and more reliable than inefficient and error-prone serial connections in linking target computers and the machines writing the software.

Around 1985 the realization struck them: With each consulting project, Fiddler and Wilner were building a portfolio of valuable intellectual property. They had created a software-development tool kit that could be reused – technology much more valuable than the applications they were writing. "When people became more interested in our tools and our operating system than in our consulting, that was an epiphany," recalls Wilner, now 44, an engineer with a degree from UC Berkeley.

"At that point," says Fiddler, "I was very conscious that what we were doing, changing how people created embedded systems, was really going to change the world."

Scratch the world's surface and you find computers. In cars they run the transmission, they operate the telephone's speed dial, they automatically adjust thermostats to match ambient heat.

The microprocessors at the heart of these and thousands of other devices are called embedded computers; embedded systems are the operating systems and applications that perform the needed tasks. Whereas a desktop OS is general purpose, an embedded system typically resides in a dedicated device – a printer, an ATM, a car – and the interface is more specific to the function it is providing. Since an embedded application is often written for different kinds of hardware, the OS must be portable to different microprocessors and system configurations. An embedded OS also needs to be scalable down to meet such constraints as minimal memory and lack of a keyboard. Desktop OSes, by contrast, tend to grow larger and more resource hungry. And, unlike PC operating systems, many embedded systems require real-time response: Hospitals expect bedside monitors to respond in a microsecond.

Embedded systems have been around since the '70s, when they first ran calculators, digital wristwatches, and *Pong*. With new, inexpensive, and more potent but less power-hungry chips promising to computerize just about anything, Fiddler and Wilner realized that their reusable code and their development process could become the perfect tool for expanding the market.

Consider antilock brakes. Today they are simple devices that pulse when you step on the pedal. But in a year or two, as they become more information driven, when you drive into a slick spot the brakes may stop the wheels on one side but speed up the opposite wheels to compensate. Or the brakes might steer the front wheels. An embedded system in the active suspension might stiffen one side and soften the other to keep the car level. It could ask the engine for more power. And it will do all of this before you've even had a chance to step on the pedal or turn the steering wheel.

To date, the vast majority of software for embedded systems has been written to order by in-house developers or contract programmers looking to fit an application to a specialized environment. With cheap and powerful CPUs, increasing bandwidth, and customer demand driving the creation of smarter machines, companies are being driven to get their products to market quickly. And, according to Sheila Ennis, a software analyst with Hambrecht & Quist, 32-bit operating systems are more difficult to develop and program than their 8-bit and 16-bit predecessors.

Hence the attractiveness of Wind River and its predeployed code – code that's been successfully running in thousands of applications for years. The company also provides high-level development tools, such as special-purpose debuggers, and other features that enable extensive testing. For the 10,000 customer products and projects in which 192 ▶

Damsels in Distress

Trained as a scientific illustrator, Swiss artist Cornelia Hesse-Honegger uses microscopes and exquisite watercolor technique to document insect mutations caused by nuclear radiation. These malformed damsel bugs were found on the grounds of a Swiss research lab where experiments were being conducted. One wing of the left bug is too short; the bug on the right displays a damaged wing, as well as a displaced and misshapen neck plate. Hesse-Honegger's gorgeous, horrifying paintings of deformed bugs are on display through September 20 at Canada's Banff Centre for the Arts; the show travels to the Vancouver Art Gallery in October. A catalog is available from Locus+ (*locusplus@newart.demon.co.uk*). – *Paulina Borsook*