

A ROBOT IN EVERY HOME

The leader of the PC revolution predicts that the next hot field will be robotics

By Bill Gates

Imagine being present at the birth of a new industry. It is an industry based on groundbreaking new technologies, wherein a handful of well-established corporations sell highly specialized devices for business use and a fast-growing number of start-up companies produce innovative toys, gadgets for hobbyists and other interesting niche products. But it is also a highly fragmented industry with few common standards or platforms. Projects are complex, progress is slow, and practical applications are relatively rare. In fact, for all the excitement and promise, no one can say with any certainty when—or even if—this industry will achieve critical mass. If it does, though, it may well change the world.

Of course, the paragraph above could be a description of the computer industry during the mid-1970s, around the time that Paul Allen and I launched Microsoft. Back then, big, expensive mainframe computers ran the back-office operations for major companies, governmental departments and other institutions. Researchers at leading universities and industrial laboratories were creating the basic building blocks that would make the information age possible. Intel had just introduced the 8080 microprocessor, and Atari was selling the popular electronic game Pong. At homegrown computer clubs, enthusiasts struggled to figure out exactly what this new technology was good for.

But what I really have in mind is something much more contemporary: the emergence of the robotics industry, which is developing

AMERICAN ROBOTIC: Although a few of the domestic robots of tomorrow may resemble the anthropomorphic machines of science fiction, a greater number are likely to be mobile peripheral devices that perform specific household tasks.

AMERICAN ROBOTIC, 1930. BY GRANT WOOD; ALL RIGHTS RESERVED BY THE ESTATE OF NAN WOOD GRAHAM; LICENSED BY VAGA, NEW YORK, N.Y., AND SUPERSTOCK, INC.; MODIFIED BY KENN BROWN



COPYRIGHT 2006 SCIENTIFIC AMERICAN, INC.

GRANT
WOOD

THE ROBOT AND THE PC CAN BE FRIENDS

Linking domestic robots to PCs could provide many benefits. An office worker, for example, could keep tabs on the security of his home, the cleaning of his floors, the folding of his laundry, and the care of his bedridden mother by monitoring a network of household robots on his desktop PC. The machines could communicate wirelessly with one another and with a home PC.

in much the same way that the computer business did 30 years ago. Think of the manufacturing robots currently used on automobile assembly lines as the equivalent of yesterday's mainframes. The industry's niche products include robotic arms that perform surgery, surveillance robots deployed in Iraq and Afghanistan that dispose of roadside bombs, and domestic robots that vacuum the floor. Electronics companies have made robotic toys that can imitate people or dogs or dinosaurs, and hobbyists are anxious to get their hands on the latest version of the Lego robotics system.

Meanwhile some of the world's best minds are trying to solve the toughest problems of robotics, such as visual recognition, navigation and machine learning. And they are succeeding. At the 2004 Defense Advanced Research Projects Agency (DARPA) Grand Challenge, a competition to produce the first robotic vehicle capable of navigating autonomously over a rugged 142-mile course through the Mojave Desert, the top competitor managed to travel just 7.4 miles before breaking down. In 2005, though, five vehicles covered the complete distance, and the race's winner did it at an average speed of 19.1 miles an hour. (In another intriguing parallel between the robotics and computer industries, DARPA also funded the work that led to the creation of Arpanet, the precursor to the Internet.)

What is more, the challenges facing the robotics industry are similar to those we tackled in computing three decades ago. Robotics companies have no standard operating software that could allow popular application programs to run in a variety of devices. The standardization of robotic processors and other hardware is limited, and very little of the programming code used in one machine can be applied to another. Whenever somebody wants to build a new robot, they usually have to start from square one.

Despite these difficulties, when I talk to people involved in robotics—from university researchers to entrepreneurs, hobbyists and high school students—the level of excitement and expectation reminds me so much of that time when Paul Allen and I looked at the convergence of new technologies and

Overview/*The Robotic Future*

- The robotics industry faces many of the same challenges that the personal computer business faced 30 years ago. Because of a lack of common standards and platforms, designers usually have to start from scratch when building their machines.
- Another challenge is enabling robots to quickly sense and react to their environments. Recent decreases in the cost of processing power and sensors are allowing researchers to tackle these problems.
- Robot builders can also take advantage of new software tools that make it easier to write programs that work with different kinds of hardware. Networks of wireless robots can tap into the power of desktop PCs to handle tasks such as visual recognition and navigation.



FLOOR-CLEANING ROBOT

**FOOD- AND MEDICINE-
DISPENSING ROBOT**

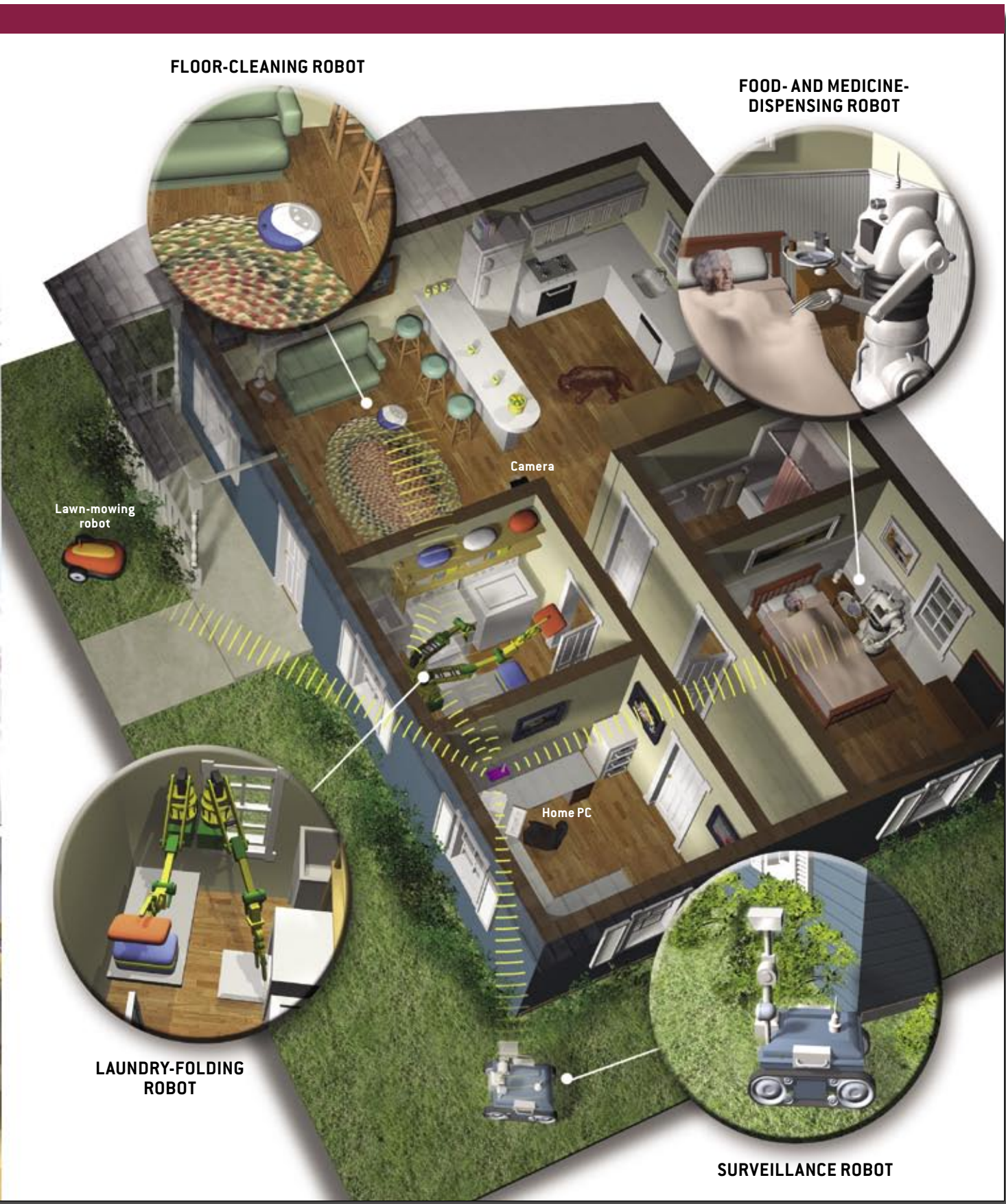
Lawn-mowing
robot

Camera

Home PC

**LAUNDRY-FOLDING
ROBOT**

SURVEILLANCE ROBOT



dreamed of the day when a computer would be on every desk and in every home. And as I look at the trends that are now starting to converge, I can envision a future in which robotic devices will become a nearly ubiquitous part of our day-to-day lives. I believe that technologies such as distributed computing, voice and visual recognition, and wireless broadband connectivity will open the door to a new generation of autonomous devices that enable computers to perform tasks in the physical world on our behalf. We may be on the verge of a new era, when the PC will get up off the desktop and allow us to see, hear, touch and manipulate objects in places where we are not physically present.

From Science Fiction to Reality

THE WORD “ROBOT” was popularized in 1921 by Czech playwright Karel Čapek, but people have envisioned creating robotlike devices for thousands of years. In Greek and Roman

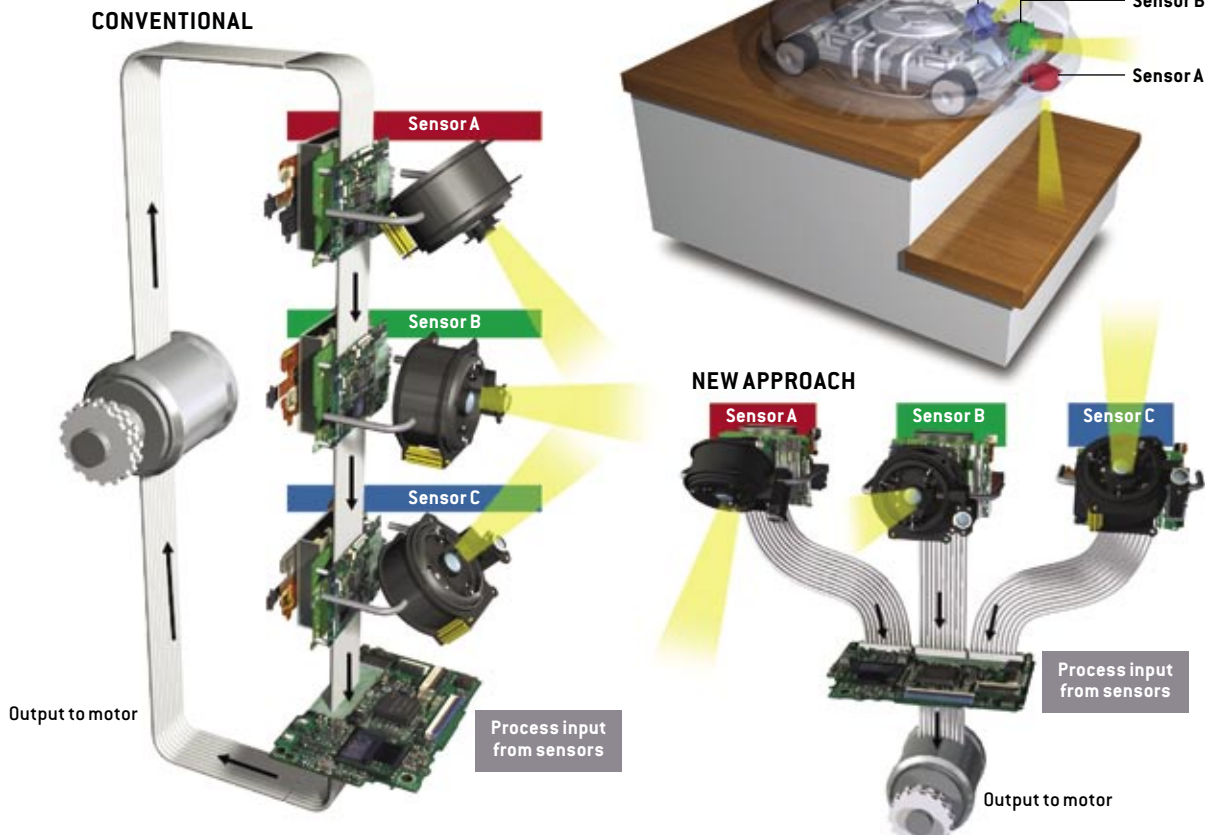
mythology, the gods of metalwork built mechanical servants made from gold. In the first century A.D., Heron of Alexandria—the great engineer credited with inventing the first steam engine—designed intriguing automatons, including one said to have the ability to talk. Leonardo da Vinci’s 1495 sketch of a mechanical knight, which could sit up and move its arms and legs, is considered to be the first plan for a humanoid robot.

Over the past century, anthropomorphic machines have become familiar figures in popular culture through books such as Isaac Asimov’s *I, Robot*, movies such as *Star Wars* and television shows such as *Star Trek*. The popularity of robots in fiction indicates that people are receptive to the idea that these machines will one day walk among us as helpers and even as companions. Nevertheless, although robots play a vital role in industries such as automobile manufacturing—where there is about one robot for every 10 workers—the fact

BETTER PROGRAMMING MEANS FEWER TUMBLES

Handling data from multiple sensors—for example, the three infrared sensors pictured on the robot at the right—can pose a dilemma. Under the conventional approach (*below*), the program first reads the data from all the sensors, then processes the input and delivers commands to the robot’s motors, before starting the loop all over again. But if sensor A (*red*) has new readings indicating that the machine is at the edge of a staircase, and the program is still processing the old

sensor data, the robot may take a nasty fall. A better approach to dealing with this problem of concurrency is to write a program with separate data paths for each sensor (*bottom right*). In this design, new readings are processed immediately, enabling the robot to hit the brakes before falling down the stairs.



is that we have a long way to go before real robots catch up with their science-fiction counterparts.

One reason for this gap is that it has been much harder than expected to enable computers and robots to sense their surrounding environment and to react quickly and accurately. It has proved extremely difficult to give robots the capabilities that humans take for granted—for example, the abilities to orient themselves with respect to the objects in a room, to respond to sounds and interpret speech, and to grasp objects of varying sizes, textures and fragility. Even something as simple as telling the difference between an open door and a window can be devilishly tricky for a robot.

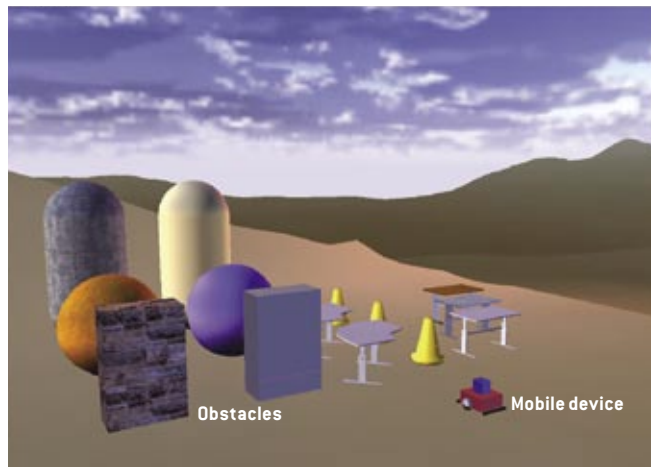
But researchers are starting to find the answers. One trend that has helped them is the increasing availability of tremendous amounts of computer power. One megahertz of processing power, which cost more than \$7,000 in 1970, can now be purchased for just pennies. The price of a megabit of storage has seen a similar decline. The access to cheap computing power has permitted scientists to work on many of the hard problems that are fundamental to making robots practical. Today, for example, voice-recognition programs can identify words quite well, but a far greater challenge will be building machines that can understand what those words mean in context. As computing capacity continues to expand, robot designers will have the processing power they need to tackle issues of ever greater complexity.

Another barrier to the development of robots has been the high cost of hardware, such as sensors that enable a robot to determine the distance to an object as well as motors and servos that allow the robot to manipulate an object with both strength and delicacy. But prices are dropping fast. Laser range finders that are used in robotics to measure distance with precision cost about \$10,000 a few years ago; today they can be purchased for about \$2,000. And new, more accurate sensors based on ultrawideband radar are available for even less.

Now robot builders can also add Global Positioning System chips, video cameras, array microphones (which are better than conventional microphones at distinguishing a voice from background noise) and a host of additional sensors for a reasonable expense. The resulting enhancement of capabilities, combined with expanded processing power and storage, allows today's robots to do things such as vacuum a room or help to defuse a roadside bomb—tasks that would have been impossible for commercially produced machines just a few years ago.

A BASIC Approach

IN FEBRUARY 2004 I visited a number of leading universities, including Carnegie Mellon University, the Massachusetts Institute of Technology, Harvard University, Cornell University and the University of Illinois, to talk about the powerful role that computers can play in solving some of society's most pressing problems. My goal was to help students understand how exciting and important computer science can be, and I hoped to encourage a few of them to think about careers in technology. At each university, after delivering my



COMPUTER TEST-DRIVE of a mobile device in a three-dimensional virtual environment helps robot builders analyze and adjust the capabilities of their designs before trying them out in the real world. Part of the Microsoft Robotics Studio software development kit, this tool simulates the effects of forces such as gravity and friction.

speech, I had the opportunity to get a firsthand look at some of the most interesting research projects in the school's computer science department. Almost without exception, I was shown at least one project that involved robotics.

At that time, my colleagues at Microsoft were also hearing from people in academia and at commercial robotics firms who wondered if our company was doing any work in robotics that might help them with their own development efforts. We were not, so we decided to take a closer look. I asked Tandy Trower, a member of my strategic staff and a 25-year Microsoft veteran, to go on an extended fact-finding mission and to speak with people across the robotics community. What he found was universal enthusiasm for the potential of robotics, along with an industry-wide desire for tools that would make development easier. "Many see the robotics industry at a technological turning point where a move to PC architecture makes more and more sense," Tandy wrote in his report to me after his fact-finding mission. "As Red Whitaker, leader of [Carnegie Mellon's] entry in the DARPA Grand Challenge, recently indicated, the hardware capability is mostly there; now the issue is getting the software right."

Back in the early days of the personal computer, we realized that we needed an ingredient that would allow all of the pioneering work to achieve critical mass, to coalesce into a

THE AUTHOR **BILL GATES** is co-founder and chairman of Microsoft, the world's largest software company. While attending Harvard University in the 1970s, Gates developed a version of the programming language BASIC for the first microcomputer, the MITS Altair. In his junior year, Gates left Harvard to devote his energies to Microsoft, the company he had begun in 1975 with his childhood friend Paul Allen. In 2000 Gates and his wife, Melinda, established the Bill & Melinda Gates Foundation, which focuses on improving health, reducing poverty and increasing access to technology around the world.



BIRTH OF AN INDUSTRY: Robot makers have so far introduced a variety of useful machines, but the designs are wildly different. Stanley (*above*), an autonomous vehicle built by the Stanford Racing Team, won the 2005 DARPA Grand Challenge, traversing more than 130 miles of desert without the aid of a human driver. iRobot, a company based in Burlington, Mass., manufactures the Packbot EOD (*opposite page*), which assists with bomb disposal in Iraq, as well as the Roomba (*right*), which vacuums hardwood floors and carpets. And Lego Mindstorms (*this page, far right*), a tool set for building and programming robots, has become the best-selling product in the history of the Lego Group, the Danish toy maker.



real industry capable of producing truly useful products on a commercial scale. What was needed, it turned out, was Microsoft BASIC. When we created this programming language in the 1970s, we provided the common foundation that enabled programs developed for one set of hardware to run on another. BASIC also made computer programming much easier, which brought more and more people into the industry. Although a great many individuals made essential contributions to the development of the personal computer, Microsoft BASIC was one of the key catalysts for the software and hardware innovations that made the PC revolution possible.

After reading Tandy's report, it seemed clear to me that before the robotics industry could make the same kind of quantum leap that the PC industry made 30 years ago, it, too, needed to find that missing ingredient. So I asked him to assemble a small team that would work with people in the robotics field to create a set of programming tools that would provide the essential plumbing so that anybody interested in robots with even the most basic understanding of computer programming could easily write robotic applications that would work with different kinds of hardware. The goal was to see if it was possible to provide the same kind of common, low-level foundation for integrating hardware and software into robot designs that Microsoft BASIC provided for computer programmers.

Tandy's robotics group has been able to draw on a number of advanced technologies developed by a team working under the direction of Craig Mundie, Microsoft's chief research and strategy officer. One such technology will help solve one of the most difficult problems facing robot designers: how to simultaneously handle all the data coming in from multiple sensors and send the appropriate commands to the robot's motors, a

challenge known as concurrency. A conventional approach is to write a traditional, single-threaded program—a long loop that first reads all the data from the sensors, then processes this input and finally delivers output that determines the robot's behavior, before starting the loop all over again. The shortcomings are obvious: if your robot has fresh sensor data indicating that the machine is at the edge of a precipice, but the program is still at the bottom of the loop calculating trajectory and telling the wheels to turn faster based on previous sensor input, there is a good chance the robot will fall down the stairs before it can process the new information.

Concurrency is a challenge that extends beyond robotics. Today as more and more applications are written for distributed networks of computers, programmers have struggled to figure out how to efficiently orchestrate code running on many different servers at the same time. And as computers with a single processor are replaced by machines with multiple processors and "multicore" processors—integrated circuits with two or more processors joined together for enhanced performance—software designers will need a new way to program desktop applications and operating systems. To fully exploit the power of processors working in parallel, the new software must deal with the problem of concurrency.

One approach to handling concurrency is to write multithreaded programs that allow data to travel along many paths. But as any developer who has written multithreaded code can tell you, this is one of the hardest tasks in programming. The answer that Craig's team has devised to the concurrency problem is something called the concurrency and coordination runtime (CCR). The CCR is a library of functions—sequences of software code that perform specific tasks—that makes it easy

GENE BLEVINS/L.A. Daily News/Corbis (Stanley); © 2006 IROBOT CORPORATION (Packbot EOD and Roomba); © 2004 THE LEGO GROUP (Lego Mindstorms)



to write multithreaded applications that can coordinate a number of simultaneous activities. Designed to help programmers take advantage of the power of multicore and multiprocessor systems, the CCR turns out to be ideal for robotics as well. By drawing on this library to write their programs, robot designers can dramatically reduce the chances that one of their creations will run into a wall because its software is too busy sending output to its wheels to read input from its sensors.

In addition to tackling the problem of concurrency, the work that Craig's team has done will also simplify the writing of distributed robotic applications through a technology called decentralized software services (DSS). DSS enables developers to create applications in which the services—the parts of the program that read a sensor, say, or control a motor—operate as separate processes that can be orchestrated in much the same way that text, images and information from several servers are aggregated on a Web page. Because DSS allows software components to run in isolation from one another, if an individual component of a robot fails, it can be shut down and restarted—or even replaced—without having to reboot the machine. Combined with broadband wireless technology, this architecture makes it easy to monitor and adjust a robot from a remote location using a Web browser.

What is more, a DSS application controlling a robotic device does not have to reside entirely on the robot itself but can be distributed across more than one computer. As a result, the robot can be a relatively inexpensive device that delegates complex processing tasks to the high-performance hardware found on today's home PCs. I believe this advance will pave the way for an entirely new class of robots that are essentially mobile, wireless peripheral devices that tap into the power of desktop PCs to handle processing-intensive tasks such as visual recognition and navigation. And because these devices can be networked together, we can expect to see the emergence of groups of robots that can work in concert to achieve goals such as mapping the seafloor or planting crops.

These technologies are a key part of Microsoft Robotics

Studio, a new software development kit built by Tandy's team. Microsoft Robotics Studio also includes tools that make it easier to create robotic applications using a wide range of programming languages. One example is a simulation tool that lets robot builders test their applications in a three-dimensional virtual environment before trying them out in the real world. Our goal for this release is to create an affordable, open platform that allows robot developers to readily integrate hardware and software into their designs.

Should We Call Them Robots?

HOW SOON WILL ROBOTS become part of our day-to-day lives? According to the International Federation of Robotics, about two million personal robots were in use around the world in 2004, and another seven million will be installed by 2008. In South Korea the Ministry of Information and Communication hopes to put a robot in every home there by 2013. The Japanese Robot Association predicts that by 2025, the personal robot industry will be worth more than \$50 billion a year worldwide, compared with about \$5 billion today.

As with the PC industry in the 1970s, it is impossible to predict exactly what applications will drive this new industry. It seems quite likely, however, that robots will play an important role in providing physical assistance and even companionship for the elderly. Robotic devices will probably help people with disabilities get around and extend the strength and endurance of soldiers, construction workers and medical professionals. Robots will maintain dangerous industrial machines, handle hazardous materials and monitor remote oil pipelines. They will enable health care workers to diagnose and treat patients who may be thousands of miles away, and they will be a central feature of security systems and search-and-rescue operations.

Although a few of the robots of tomorrow may resemble the anthropomorphic devices seen in *Star Wars*, most will look nothing like the humanoid C-3PO. In fact, as mobile peripheral devices become more and more common, it may be increasingly difficult to say exactly what a robot is. Because the new machines will be so specialized and ubiquitous—and look so little like the two-legged automatons of science fiction—we probably will not even call them robots. But as these devices become affordable to consumers, they could have just as profound an impact on the way we work, communicate, learn and entertain ourselves as the PC has had over the past 30 years. SA

MORE TO EXPLORE

More information about robotics in general is available at:

Center for Innovative Robotics: www.cir.ri.cmu.edu

DARPA Grand Challenge: www.darpa.mil/grandchallenge/

International Federation of Robotics: www.ifr.org

The Robotics Alliance Project: www.robotics.nasa.gov

Robotics Industries Association: www.roboticonline.com

The Robotics Institute: www.ri.cmu.edu

The Tech Museum: Robotics: www.thetech.org/robotics/

Technical details and other information about Microsoft Robotics Studio can be found at msdn.microsoft.com/robotics