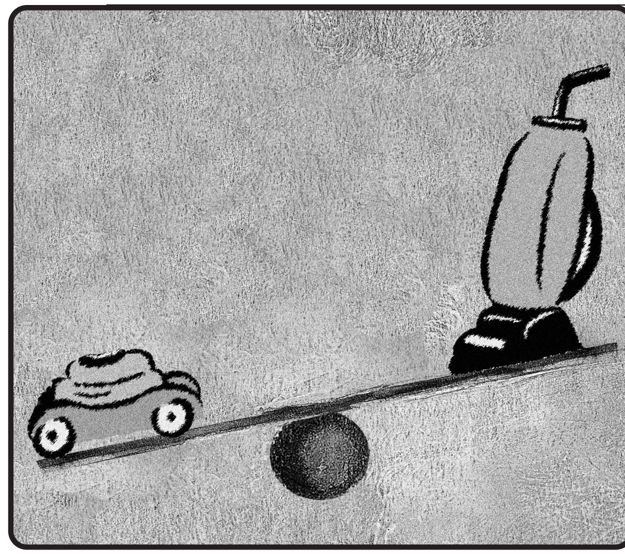
Robots at the Tipping Point



© EYEWIRE, ARTVILLE

The Road to the iRobot Roomba

BY JOSEPH L. JONES

The first IEEE-IFR joint forum on Innovation and Entrepreneurship in Robotics and Automation took place on April 20th 2005 in Barcelona, Spain. The following article was presented at the forum as one of the three nominees selected for the Innovation and Entrepreneurship in Robotics and Automation Award. For further information, please visit http://teamster.usc.edu/~iera05/index.htm.

—Hadi Moradi
General Secretary
Industrial Activity Board

Consumer robotics has achieved its first significant commercial success with the arrival of the iRobot Roomba Robotic Floorvac. Since its launch in 2002, over 1.2 million Roombas have entered service in homes all over the world (see Figure 1). Roomba accomplished this singular feat largely by exploiting one simple idea: A robot's cost must be commensurate with its utility.

History

Roomba's earliest direct ancestor emerged from the Massachusetts Institute of Technology (MIT) Artificial Intelligence (AI) Laboratory in 1989. In January of that year, Prof. Rodney Brooks' Mobile Robot Group staged an event known as the AI Olympics. The group handed out kits of parts—microprocessors, sensors, motors, LEGO pieces, and so on—to all interested parties at the laboratory. Approximately 60 individuals, singly and in groups, set about building robots.

My own efforts in the AI Olympics resulted in the floor-sweeping robot, Rug Warrior, shown in Figure 2. Although Rug Warrior lacked the robustness and reliability necessary for a consumer product, it provided a promising demonstration that a small, uncomplicated device might be able to clean floors.

As the construction of Rug Warrior suggests, all the components Roomba needed were available in 1989, albeit at prices somewhat higher than today's. Funding to develop a product, however, was not available. This key element remained stubbornly absent for the next ten years.

In 1999, a colleague, Paul Sandin, and I proposed to iRobot that we investigate building a floor-cleaning robot targeted at consumers. Developing a small floor-cleaning robot was appealing because, at least anecdotally, a market for such a product seemed to exist. Several iRobot employees observed that, after being introduced as a member of a robot company, very often the next question asked would be: "Can you build a robot that will clean my floors?"

The company accepted our proposal to build a proof-of-concept floor-cleaning robot and, with two weeks of focused work, we produced the robot shown in Figure 3. This robot, called Scamp, employed a carpet sweeper cleaning mechanism and a force-sensing bumper. It could anticipate collisions using several infrared emitter/detector pairs mounted on the skirt.

Although its abilities were rudimentary, the company saw great promise in Scamp. iRobot supported this view by forming a team to develop the rough prototype into a viable consumer product. Team composition varied over time, but, for the most part, the Roomba team consisted of five engineers, a manager, a marketer, and an assistant.

Technology

The robot we ultimately developed relied only on inexpensive, standard components but often used these components in creative ways. To date, five patents have been issued on aspects of Roomba.

Roomba's shape is round to maximize the robot's ability to escape from obstacles—a round robot can always turn in place. The power system uses standard rechargeable batteries. Because of Roomba's energy-efficient cleaning mechanism, battery capacity is sufficient for 1–2 h of running time. The robot's microprocessor is a modest one, possessing a clock speed of 16

1070-9932/06/$20.00©2006 IEEE

MHz, 256 B of RAM, and about 30 input/output (I/O) lines.

Roomba's sensors include a 2-b, front-mounted bump sensor, cliff and wall sensors, a room confinement sensor (Roomba will not cross a user-positioned infrared beam), drive wheel-mounted shaft encoders, and a battery voltage sensor. Newer models also include dirt sensors and a stasis sensor (triggered when the robot tries but fails to move).

A total of five motors are incorporated into Roomba's actuators. Two motors power the differential drive mobility system, and one motor each is devoted to the main brush, the side brush, and the vacuum.

A behavior-based programming scheme directs Roomba. The program contains many highly tuned strategies for avoiding and escaping tight spots, stuck situations that don't trigger the bump sensor, and other hazards. Roomba's cleaning strategy, combining random bounce and wall following, is designed to maximally cover the floor, even in the presence of clutter.

Figure 1. Four models of iRobot's Roomba Robotic Floorvac are currently available.

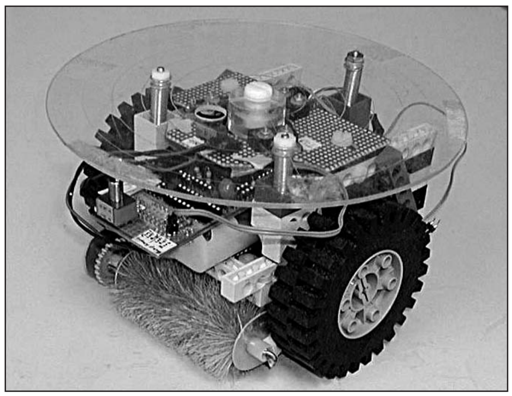


Figure 2. Rug Warrior was an early floor-cleaning robot. It possessed a simple carpet sweeping mechanism for cleaning, a full coverage bump sensor to allow escape from collisions, and drive wheel-mounted shaft encoders that enabled the robot to follow a simple cleaning pattern. A behavior-based program running on a low-cost microprocessor controlled the robot.

Figure 3. The first iRobot Roomba prototype was known as Scamp. Here Scamp is shown on its display pedestal. The robot has a vacuum-formed shell and a switch that allows the user to select among three cleaning modes: wall-following, spiral-pattern, or random bounce.

Principles

Development of the initial Roomba model took approximately three years. During this period, several important principles guided our labors:

1) The application comes first.
2) Cost matters.
3) Only real-world testing can reveal the robot's flaws.
4) "Usually" is unreliable.
5) Complexity kills.

We chose these principles, in large measure, as a reaction to problems we observed in other commercial robotic developments.

Application

Before Roomba, robots were popularly imagined as futuristic, high-tech, machines able to perform a wide variety of functions. Star Wars' R2-D2 and the robots from Isaac Asimov's stories are archetypes of this view. Even many robot designers implicitly accepted this vision. When asked, "what does your robot do?," such designers answer, "everything, whatever the user programs it to do." But customers balk at purchasing an "everything" robot, correctly interpreting the answer to mean, "nothing, fresh out of its packing box the robot will perform no useful functions."

The iRobot Roomba team concluded that for a robot to be successful, the question "what does the robot do?" must have a clear and compelling answer; that is, we, the designers, needed to settle on a specific application (or applications) rather than burden the consumer with this chore. We further needed to optimize the robot to perform its function effectively while requiring only minimal attention on the part of the user. From the customer's point of view it is the application, not the robot, that matters most.

Roomba cleans floors. This concise statement answers the question, "what does the robot do?" Given this declaration, customers can easily understand how the robot will benefit them, and how much this benefit might be worth. Furthermore, such a clear description of the application simplifies development-phase decision making. Should the issue arise, for example, it is clearly preferable to assign scarce resources to enhance the robot's cleaning mechanism than to, say, add a touch screen to the robot's user interface. The former helps the robot clean floors; the latter does not.

Cost

In the marketplace, a robot must compete with every device or service provider that offers a function similar to the robot's function. Consumers reasonably compare the costs of accomplishing a task in different ways. Roomba competes with familiar items—vacuum cleaners and cleaning services—and customers know the price of a clean floor. Were the cost of cleaning a floor using Roomba not competitive with the cost of other forms of cleaning, Roomba would attract few customers. Customers who did pay a too-large price might do so for the cachet of owning a robot but not because they had found a more advantageous way to clean their floors.

Thus, cost was one of our most important considerations in designing Roomba. Every feature we contemplated incorporating into Roomba was measured against the cost it would add to the robot and, therefore, impose on our customers. Only features that improved the robot's ability to do its job in an essential way were included in the robot.

Testing

During the development of Roomba a certain pattern became very familiar to the developers. We would assemble a prototype robot, test it in our lab, discover several problems, fix all the problems, and then verify that the robot worked by retesting. Next, we would operate the robot in the home of an iRobot employee. Invariably some particular feature in that new environment caused the robot not to work effectively; typically the robot would become stuck. The robot would go back to the lab for analysis and redesign. Next the robot would return to the employee's home to verify that we had indeed remedied the problem. When the robot worked there, we would test it in another home. Here the robot would discover yet another situation that caused it to function improperly and the cycle would begin again.

Neither simulation nor careful thought could substitute for extensive testing. Even seemingly benign indoor spaces are rich in detail that defies simple models. The floor is not flat, the edge of a step is not square, and traction is not always sufficient. We learned these and numerous other painful lessons repeatedly during testing. Had our development taken place only in the laboratory, Roomba would have been ill-equipped to meet the rigors of the real world.

"Usually"

Given a tight development schedule and limited resources, there is always a temptation simply to ignore robot-challenging situations that we expect will not "usually" occur. Unfortunately, this strategy is counterproductive when our goal is to maximize the probability that the robot will complete its task. The many iterations (described in the previous section) required to achieve a design effective in a variety of environments illustrate the point.

The failure of the robot to complete its task (by, for example, getting stuck) tends to be systematic rather than random. If the environment contains even a single unplanned-for hazard, the robot will almost invariably encounter the hazard and become stuck; if all such hazards are absent, the robot will function properly. This observation had the effect of greatly increasing our sensitivity to perceived problems. We often chose to design into the robot a solution to an anticipated problem, even when we had never seen a robot succumb to that particular problem during testing. This provided some insurance that the robot would operate properly in environments with which we had no experience.

Complexity

Besides the cost issue mentioned above, a second reason to exclude nonessential systems from the robot is to minimize complexity. Complexity has a great capacity to kill budgets, schedules, and ultimately products. Robots are especially susceptible to complexity-induced difficulties. This is because the field is immature, meaning that few complexity-managing heuristics are yet in place, and also because the robot's various systems often interact in unexpected ways. For example, during development, one model of Roomba stopped homing on its recharging station when the color of the bumper was changed. Such unanticipated interactions between systems take time, and often, extra cost to correct. The fewer systems a robot possesses the fewer surprises developers will face. Because of this, it is sometimes better to implement a needed feature by inventing a simple, new system than to add two or more familiar systems to the robot to accomplish the same purpose.

Summary

Roomba has succeeded in the marketplace where other robots failed or feared to tread. Like many good products, Roomba has been rewarded for putting customer's interests first. Roomba accomplishes a task customers care about and does so at a price competitive with other methods. No breakthrough technologies were required, just familiar components used in new ways. A simple set of principles helped designers provide value to our customers by keeping the focus on the application, robot cost and functionality, and system complexity.

Keywords

Behavior based, consumer robot, floor cleaning, mobile robot.

Joseph L. Jones is a senior roboticist at iRobot Corporation in Burlington, Massachusetts, and has been with the company for 13 years. His primary interest is the practical application of robotic technology to real-world problems. Prior to joining iRobot, Mr. Jones served on the research staff of the Massachusetts Institute of Technology (MIT) Artificial Intelligence Laboratory for nine years. Earlier he studied physics at MIT, earning an S.B. in 1975 and an S.M. in 1978. Jones is an author of three books on robotics and is the designer of a popular robot kit. He grew up in a small rural community in the Missouri Ozarks.

Address for Correspondence: Joseph L. Jones, iRobot Corporation, 63 South Avenue, Burlington, MA 01803 USA. Phone: +1 781 418 3246. Fax: +1 781 345 0201. E-mail: jlj@irobot.com.