Chapter 6

Conclusion and Future Directions

This dissertation has been concerned with the relationship between two central questions:

- How does one design an evocative learning environment (based on robotic design) for undergraduate-level students?
- What do students do when confronted with this learning environment?

As suggested by Figure 6-1, the meanings of these two questions is defined by exploring the relationship between them. The learning environment presented to our students, as described in Chapter 3, consists jointly of the physical hardware and software platform, the structure of the contest challenge, and the pedagogical organization of the Robot Design course. Through four successive iterations of our project, this dissertation has presented the data of our work in creating environments for robotic design, and the students' work of using these materials to build robot projects. This dissertation has presented a set of interrelated issues that have been studied: our learning how to build robots, our learning how to design materials for robot-building, and our learning what students are learning when they build robots. Thus the thesis has not only documented the students' experiences, but our own learning process in developing the Robot Design environment.

In analyzing students' approaches to control, we saw a strong tendency for building lock-step, algorithmic processes that expected or required precise and accurate sensory readings. Students expected the sensor data to provide them with unambiguous information for interpreting their robot's relationship to its environment. The approach toward control

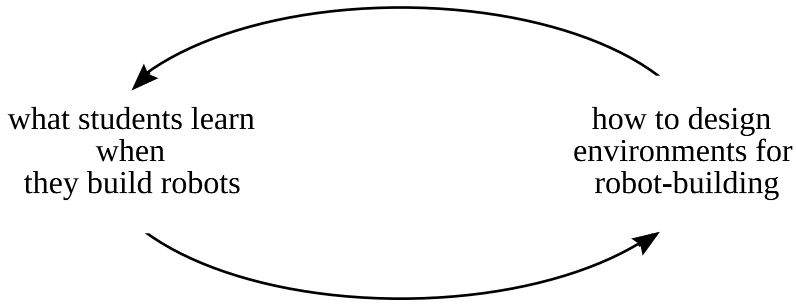

Figure 6-1: Designing robot-building environments and studying learners' experiences with them

required for a robust robot was unfamiliar and uncomfortable to the students. Many students interpreted their robots' failures by pointing to particular components of the robot which had "failed" rather than the architecture of their overall solutions, which were fundamentally vulnerable to such failures. I conclude that the robot design experience presented students with a problem that was strongly challenged their notions of systems and control. This sort of experience is necessary in order for students to reconsider their deeply held views.

This study has also revealed a disparateness between phenomena that students observe during their robot-design processes and the formal models of engineering science that may be used to explain such phenomena. When students are confronted with puzzling situations during their design activities, they cannot readily produce the engineering science that may be used to explain the phenomena they observe. This happens even though it may be demonstrated that students "possess" the requisite knowledge in other contexts. Teaching students theory in traditional academic contexts does not means they will have access to this knowledge in design-oriented settings; yet, as fledgling engineers, design situations are precisely what they will face. Therefore I conclude that design experiences like those encountered by the students of the Robot Design should be a fundamental part of the undergraduate curriculum.

In this conclusion, I revisit the central themes from the three main chapters of this dissertation. Interwoven with these summaries are suggestions for future directions of work, and preliminary results based on the influence of this work on the larger community of robotics researchers and university educators. First, within the context of the design of our educational technology, I discuss the level of abstraction of the materials provided to students, including the approach we settled on, choices made by other educators for their own courses, and the potential for developing robot-design materials for secondary education. Next, I return to the issue of students' approaches to the problem of control, and suggest a way of modifying the robot contest specification to further the valuable challenges provided to the students. Finally, I revisit the discussion on design styles, and argue that irrespective of students' approach to the design problem, they encounter the difficulty of reconciling the phenomena they encounter with their formal engineering design knowledge.

6.1 Educational Technology

Chapter 3, "Technology for Learning," explored a set of issues related to the design and assessment of an educational technology—specifically, a technology optimized for the construction of small mobile robots and the exploration of issues related to the design of the same. We examined this technology from several perspectives, including the design of the robotic environments (the contest specifications), the design of a hardware and software platform to support students' robotic projects, and the creation of sensor devices to enrich the robots' capabilities and the students' potential for learning.

The problem of designing effective contests involved a number of issues. The contests had to focus students on task of creating a viable robotic device, but at the same time, provide freedom within the constraints, and encourage a multiplicity of solution strategies. One of the primary goals was to impart the attitude that there are lots of ways to solve a problem, not one necessarily right one.

Through the iterative process of designing and evaluating contests and course versions, we learned the structure of the "problem of the problem,"[1] that is, the challenge of designing and operating the project. The contests, hardware and software, and sensor devices were all part of the overall problem of developing, structuring and administrating the class; of creating an environment that is rich in learning opportunities, provided enough support that students feel able to take charge of their own experience, but did not limit them or constrain them from diving into pockets of interest that arise. In a large part, the chapter was a thick

[1]This phrase was suggested by Donald Schön.

description of the issues that we faced as course designers, how we dealt with them, and my own interpretations of the resulting effects on students' learning.

6.1.1 Levels of Abstraction

A turning point in the evolution of the project occurred when we decided to create the robot programming environment based on the Interactive C language, which replaced the earlier assembly language methods. By providing students with a higher level of abstraction for interacting with the robot hardware, we allowed them to focus on issues of strategy and algorithm rather than details of registers and bitmasks, and to create much more sophisticated robot systems. The alternate choice of providing the lower level of microprocessor programming would be equally valid given different pedagogical goals.

The development of the Interactive C system was partly motivated problems we observed students experiencing with the assembly language system. The hardware and software of the assembly language materials suffered from quite poor *observability*: when something went wrong, it was very difficult to tell at which level of the system the problem lay. In other words, it was difficult to ascertain whether a perceived difficulty was due to a catastrophic hardware failure, a hardware problem in a sensor, an algorithmic problem in interpreting a sensor's data, or a software problem in the algorithm for interpreting a sensor's data. As described in Chapter 3, this commingling of possible failure modes caused a great deal of frustration among the students.

By providing students with tools to discriminate among these failure modes, the Interactive C system was a great step forward. But it did much more than this. It allowed students to express algorithmic ideas in a familiar and more readable form than assembly language; it allowed them to create more complex programs (which facilitated the students explorations into the control issues discussed in Chapter 4). This expressive power, however, came at a price: students were insulated from a variety of the details of their robots' operation. Some degree of abstraction is part of any project; one does not discard the infrastructure of technology at every turn. The question we faced was whether the abstraction we had introduced was a worthwhile tradeoff.

When we were developing the Interactive C system, part of our justification was that it would solve the observability problems we had experienced with the assembly language system. Most of these problems, however, could have been addressed while preserving the assembly language system. For example, the "system heartbeat" indicator, which revealed a catastrophic hardware problem at a glance, could have been installed on the assembly language system. Similarly, code libraries could have provided the capability to display on sensor values on the computer console in a similar manner as did the LCD display on the Interactive C system.

Nevertheless, we could not have created the degree of interactivity that we did with Interactive C with the assembly language materials. While the level of abstraction matter can be argued in either direction—depending on one's educational objectives, sometimes it's better to give more expressive power; sometimes it's better to keep students close to the level of the hardware—the interactivity of the C language system gave students the chance to rapidly try out their programming ideas, create more sophisticated programs, and explore issues of sensing and control in greater depth.

6.1.2 Robot Design at Other Universities

After the end of the 1991 course, we realized that the robot-building technology we had created could be of value to others who were interested in robots, both for educational purposes (as in our project) and for research purposes. By 1992, we arranged with the MIT administration for free distribution of the course technology, including the controller boards, the custom Interactive C software, and the 250-page course notes (Martin, 1992). We also released a revised version of the more simple 1990 hardware, which we called the "Mini Board" (Martin, 1993).

In addition, the project received publicity through popular media exposure in monthly magazines (Arneke, 1990; Freedman, 1990) and daily newspapers (Chandler, 1992), research publications (Martin & Sargent, 1992; Martin, 1992, 1993), and via Internet newsgroups (the electronic equivalent of "word of mouth"). An eclectic community comprised of robotic hobbyists, academic educators and researchers, and industrial professionals came into being, brought and held together largely by the existence of electronic communication (i.e., e-mail). Our technology was adopted by these users it because of its unique packaging of features suitable for robot design projects, its ease of use, and, more recently, the existence of a community of supportive users.[2]

The process by which this community arose and now flourishes is a matter of separate interest (it would not have happened without the widespread adoption of electronic mail within the professional and academic world), but here it is valuable to note the ways in which the materials have been employed by others for a variety of purposes. I have observed three broad categories of usage: educational initiatives, research projects, and hobbyist/professional purposes.

The choice we faced in choosing the higher-level Interactive C environment has also presented itself to course developers at a number of universities who have started courses inspired by the MIT Robot Design project. Many of the courses others have created share the central activity in which students design their own little robots, but they have different specific pedagogical goals which depend on the interests of the various course developers. This illustrates both the adaptivity of the technology we designed, and the creativity of university educators in building their own learning environments.

At the time of this writing, I estimate that there are between twenty and thirty such projects. Here I present just a few.

A microcomputer architecture course. At the New Mexico State University (NMSU), Professors Patricia Teller and Joseph J. Pfeiffer, Jr. are using a version of the technology developed for the Robot Design project in their sophomore-level class on microcomputer architectures. Their class is built around the activity of designing and programming a small robot built from LEGO parts and operated by a microprocessor controller.

In the NMSU class, students program in assembly language. This was a deliberate decision by the course designers because they wanted students to learn about the inner structure of microprocessors. Professors Teller and Pfeiffer report that because of the hands-on, physical action of the robots' motors and sensors, students have been more

[2]This observation is based on an informal survey I recently conducted of members of this electronic community.

successful in learning assembly language than in previous courses.[3]

A senior design course. At Bucknell University, Professor Thomas Sloane has been running a senior design course based on a robot design contest since 1991, inspired by our *Robo-Pong* contest. In his version of the design project, however, students design the microprocessor control hardware in addition to building robots to solve the contest problem. As he describes the project, “different systems come and go depending on the participants’ goals.”[4] Thus, in the Bucknell course, students learn digital design as a part of their robotic projects.

A programming course for mechanical engineering students. At the University of Wollongong in Australia, Peter Dunster has introduced hands-on robotics projects within the Mechanical Engineering department. He was motivated by a “woeful lack of programming knowledge and almost no practical experience” in his final year students. With the support of a department head also looking to bring more practical subjects into the syllabus, he has established a successful project course.[5]

An undergraduate communications course. At the Trondheim College of Engineering in Norway, Professor Fredrik Wilhelmsen is using our technology with undergraduate students to teach data communications and distributed systems. One project involves simple interacting robots, and another employs a controller board operating a video camera through an infrared link. Professor Wilhelmsen cites the need to broaden students’ understanding of the potential of computers: “The nice thing about the Mini Board is that it makes the real world easily accessible from the computer. It is very easy to connect sensors and actuators. . . Most students think that the screen and the printer is the only way to get [information] out of a computer.”[6]

An introductory graduate course in robotics. In the Computer Science and Engineering department at the University of Connecticut, Professor Robert McCartney is

[3] Personal communication, 1993.
[4] Personal communication, 1993.
[5] Personal communication, 1993.
[6] Personal communication, 1994.

offering a graduate course (spring 1994 semester) in which students build their own robots on various mechanical platforms (modified toy cars and simple three-wheel bases) and use the same control technology developed for the MIT Robot Design project. In Professor McCartney's course, the emphasis is on designing robots that can wander around an office building and construct a map of the floor plan. By providing students with materials to build their own robots, and a problem in contemporary robotics research, he hopes to combine the practical with the theoretical.[7]

In each of these examples, the individuals involved have adapted the technology to their own idea of what is important for students to learn. A common concern is providing practical experience to students, and a belief that learning-by-designing is valuable. Beyond this there is variation with respect to the issues discussed in Chapter 3: the level of abstraction, modularity, and structure of problem posed to the students.

6.1.3 Robot Design in Secondary Education

Many educators agree that children should be "technologically literate," but there is little consensus as to what this means in practice. While there are exceptions, typical attempts at creating technology classes suffer the same problems as the most of school: textbook-based and traditional laboratory approaches which do not facilitate an empowering relationship with technology. The Robot Design concept would allow younger students to explore technological ideas in a way that lets them reflect on their own creations and realize the significance of systems thinking in a first-hand way.

The *LEGO tc logo* materials discussed in the Background chapter have been marketed by the LEGO company for use by middle school (aged 11 through 14) children; LEGO Dacta USA has recently announced a substantial revision of this product, called *LEGO Control Lab B*, which is aimed at high school children. A similar repackaging of our technology, originally developed for university use, would allow younger learners to explore ideas of sensing, control, and engineering design.

This work to adapt the materials created for the Robot Design project for use by high

[7] Personal communication, 1994.

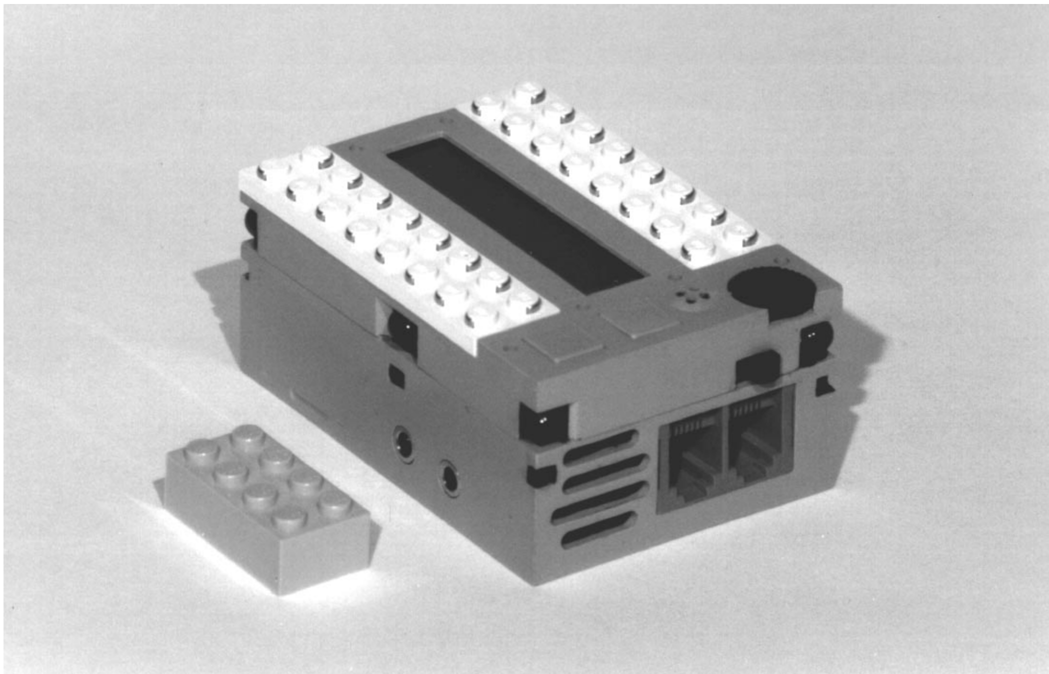

Figure 6-2: The 1994 version of the Programmable Brick

school students has already begun. In the Epistemology and Learning Group at the MIT Media Laboratory, Randy Sargent has led the design of the next generation of the LEGO Programmable Brick (shown in Figure 6-2). Building on experience gained in the first Programmable Brick work (discussed in the Background chapter), the Programmable Brick puts all of the electronics in a small plastic box and provides nice connectors and an easy to use programming environment. As one example application for the Programmable Brick, Sargent plans activities in which children will explore the concept of embedding computationally active elements in the environment (sometimes referred to as *ubiquitous computing*), allowing them to create, for example, a room that will detect when a person enters it, turning on a light or playing a greeting message (Sargent & Resnick, 1993).

With the Programmable Brick or similar technology, it is feasible to develop curriculum plans and project structures to give high school students the chance to build competitive robots like the ones that have been developed by the MIT students in this study. Whether or not they intend to pursue careers in science and technology, children deserve the chance to explore ideas of sensing, control, and engineering early in their academic careers, and in a way that encourages and develops their own creativity and sense of mastery. It is unfortunate that many school activities relating to mathematics and science have the effect of deterring children from these fields of inquiry. If children are given the chance to engage in personally meaningful design projects with a rich technological content, the result might be very different.

6.2 Models of Control

Chapter 4, "Ideal and Real Systems," explored students' notions of systems and control as instantiated in their robot designs. The picture that emerged from this study suggests that students' pre-existing notions of control led them into trouble when they approached the robot design task. Students had a great deal of difficulty understanding the properties—and limitations—of the robotic sensor devices, and more generally, the interface between the computation and the real world. It became apparent that students were accustomed to constructing algorithms that operated on precisely known quantities, as in an algorithm to sort items in a list. Most students constructed robots whose solution algorithm was quite fragile with respect to unanticipated situations, which in practice happened with disappointing frequency. This caused some students to question their models of control, but most simply chalked up the failures as being due to "Murphy's Law" or limitations in their ability to execute a design which they still believed to be correct. For many, however, a seed of doubt has been planted, and I expect these students to be less surprised in the future as they discover that the real world is much more complicated than the clean models of it that they have been taught.

The problems given to the students in the Robot Design contests brought particular control problems into focus. In general, the contests contained sub-tasks that were amenable to local, negative feedback control (e.g., climbing a hill, following a line, or following a wall) within the context of overall tasks that involved a fair bit of uncertainty (e.g., the actions of the opponent robot(s)). While some students avoided the local, low-level feedback approach in favor of attempts to centralize control through predictability, most realized the utility of the local feedback techniques. Few students, however, created robots that dealt with the uncertainty of the overall robot behavior problem.

An interesting issue for further study is the question of how closely this sort of engineering problem corresponds with the types of scenarios engineers encounter in the real world. Certainly the concept of negative feedback is central to real-world control systems, but it would be fascinating to study the correlation between vicissitudes in the world of the students' robots and the world of installed controls. When engineers build anti-lock braking systems for cars, do the sensors perform as erratically as those on the students' robots? Probably not. On the other hand, critical early warning radar defense systems have fallen prey to various forms of sensor misinterpretation (e.g., unanticipated radar reflections from the moon after more powerful radar was installed were interpreted as a missile attack by automated hardware/software systems) (Roberts & Berlin, 1987)). The lessons about the dangers of overly trusting sensors are quite apropos to the challenges of contemporary engineering.

6.2.1 Artificial Life and Behavioral Robotics

In fairness to the students of the Robot Design project, the contest specifications and the Interactive C language conspired to make it difficult for students to create robust solutions. Because contests were so short, between thirty and sixty seconds, winning robots had to go about their task quickly and without making any missteps. And because Interactive C is a traditional procedural language, it tended to encourage sequentially organized programs.

A yet more challenging robot design contest would require robots to perform a task for a significantly longer period of time (e.g., ten minutes). In this kind of contest, students would be forced to think about the control task in an entirely new way, along the lines of recent work known as *artificial life* and *behavior-based robotics*. In this work, synthetic entities (e.g., mobile robots) have very different control architectures that allow them to "survive" in unstructured environment for extended periods of time. Translated to the sort of robot design tasks that have been discussed in this dissertation, these robots would have to navigate freely around a playing field, recover from collisions with other robots, and perform some kind of task that would require a longer period of time to complete.

This kind of contest might not have the high-action thrills that characterized the previous contest challenges, but it would require students to construct robots that used behavioral responses to environmental stimuli rather than robots that ran a pre-designed specific action loop. Along with the change in contest task, it would be appropriate to introduce a control language along the lines of the activation networks suggested by Pattie Maes (Maes, 1990) and the layered behavior mechanism suggested by Rodney Brooks (Brooks, 1986). Such a contest would bring these alternate models of control to the forefront of students' attention, and would provide the structural motivation for them to explore these models.

6.2.2 The Role of Sensing

Because sensors are a robot's interface to the real world, the types of sensors that are available have a large influence on a robot's potential abilities and the richness of the robot control problem for the student. Many of the issues that students grappled with during their project work was related to a process of coming to understand the actual (as compared to ideal) functioning of a particular sensor device.

For this reason, the area of sensor development is particularly ripe for making improvements to the technology of the Robot Design project. Appendix D.4 contains a discussion of our process of developing the infrared robot-sensing technology; the use of this sensor, as was the case with many others, led to a variety of problems that were quite valuable from a standpoint of students' and our own learning.

It so happens that all of the sensors we have used are fundamentally simple devices that provide data that is mostly local in nature. A more powerful microprocessor unit, however, would provide the foundation for highly data intensive sensors—specifically, a electronic camera to give robots visual capability. The development and inclusion of such technology into the Robot Design concept would allow at least two different directions of project work:

Sophisticated robot strategies. If the vision system were supplied with predesigned functions for its use (as was the case with our other sensors), students could make use of the vision system without a detailed understanding of machine vision principles, allowing more complicated game designs and students to create more sophisticated robots.

Hands-on machine vision projects. Such a system would also allow students to explore the problems in machine vision in a hands-on way. The subject of machine vision is often taught via conventional means, since it's not typical to give each student his or her own robot, camera system, and algorithm development platform. But a hardware, software, and workshop approach that did specifically do this is a logical extension of the work developed in this thesis.

As miniature camera technology continues to drop in price and become easier to use, this concept will become eminently practical.

6.3 Design Styles and Engineering Knowledge

The crux of the problem of engineering education is helping students bridge the gap between the formal engineering science knowledge and the observed phenomena of a design situation. In their robot-building work, students were frequently confronted with confusing situations that stemmed from their use of incorrect, or lacking, models of the underlying engineering phenomena. In this type of problem-solving, the challenge is to figure out what the problem is; i.e., to come to understand what type of explanation underlies the phenomena being observed. This is a very different sort of problem-solving—it is more a *problem-finding*—than is fostered by traditional academic contexts. Students need to be given these intellectual opportunities, since in their work as professional engineers and designers, the hard part of solving a problem or doing a design is figuring out the right questions to ask to get a handle on an underconstrained situation.

For the purposes of this conclusion, I would like to reinterpret one of the events discussed in the earlier chapters (the motor driver conundrum discussed in Chapter 5) and offer some additional thoughts on the role of the teacher in facilitating students to reflect on the lessons of their robot-building experiences.

6.3.1 Mysterious Phenomena

The motor driver situation was the result of students' reaction to what they perceived to be a frustration with the stock motor electronics that they were provided in the *Robo-*

Pong year. Students built assemblies that operated adequately when controlled from the manually-operated motor switch board, but then performed poorly when driven from the stock electronic circuit. A number of students then chose the "design move" of developing their own control circuits to provide improved performance.

This activity was fraught with problems. The first was in constructing control circuits that did not violate the parts count limitation we had imposed on custom designs. The teams that did satisfy this constraint then discovered the surprising problem that their motor power output became highly dependent on the battery charge level.

In retrospect, an analysis of the situation revealed an electrical problem in which the motor power level was governed by the equation $P = V^2/R$, where P is the power level of the motor, V is the voltage of the battery, and R is the effective resistance of the control circuit. In their custom circuits, the students had greatly reduced the value of R in order to achieve greater power, and had inadvertently made the actual power level much more sensitive to changes in V.

At the time, neither we nor the students who had build the circuits were able to make this kind of analysis of the situation. It was clear that the performance problem was related to the new circuitry the students had installed, but we had not predicted it nor did we have a formal explanation for it. The situation was particularly confusing because other students who had *not* modified their motor circuits experienced a similar problem (high sensitivity to battery voltage). It turned out that there was a particular configuration of the gear train that would also cause the same effect.

(It is worth noting that there may be other interpretations or explanations for the battery voltage sensitivity problem; here I have not actually justified why I believe a $P = V^2/R$ relationship to be governing the situation. The point, however, is not which explanation is correct, but rather the process by which we come to believe one over the other. We want students to be making these kinds of arguments as part of their process of learning to become an engineer.)

The overall situation was a significant drain on our resources for assisting students, and the result was that the students who had modified their driver circuits did not achieve the overall performance benefits they had anticipated. While their robots were indeed more powerful, the students were unable to harness this power in their programming. We had not yet figured out the technical reasons behind the battery sensitivity problem, and our response to the situation was to *steer students clear of the problem*. In the subsequent year, we made three changes in this regard: (1) we deliberately "crippled" the manual motor switch boards so that students wouldn't be tempted by seeing a power level that they wouldn't be able to achieve in their electronic control; (2) we incrementally improved the performance of the electronic control; and (3) we disallowed modifications to the stock electronic circuit.

I was not altogether pleased with this solution at the time, and I consider it even less desirable in retrospect. The motor driver experiments were a rich source of learning for those students who participated in them; while we did reduce the need to explore this path, we also unequivocally shut it down. With our current ability to explain the phenomena that were mysterious at the time, perhaps we could re-open this area for interested students to explore.

6.3.2 The Role of Reflection

Perhaps a shortcoming of the Robot Design project as an educational venue is the fact that there is no formal opportunity for students to reflect on their experiences after the contest itself; due to time constraints, the contest is the last gathering for course participants (with the exception of students who choose to prepare their robots for the rematch competition held at the Boston Computer Museum a few months later).

The educational challenge is to find the best forum or circumstances for students to consider the lessons of their design work. In the "exit interviews" we conducted with students a few days before the contest, I was struck by the degree to which students had difficulty talking about their experiences in the course. Students who were quite involved and animated during the design process often had little to say about what had been meaningful to them or what they had "learned."

Further, students found it difficult to abandon their deeply held views about systems. Even after seeing robot after robot fail during the contest, most students would complain of a particular sensor or mechanism that had failed on their robot rather than consider the possibility that the overall approach toward control they had taken was fundamentally susceptible to component failure.

Experiences like the Robot Design project should not be isolated events in our educational systems, and there's certainly no reason they should be restricted to a university setting. The best way to make the theoretical knowledge that we consider important valuable to a student is to give him or her a chance to put it to use. When we discover the gap of "messiness" that lies between theory and practical applications, we should not ignore it or toss it away as uninteresting, but encourage ourselves and our students to dive in and explore the complexity of putting ideas into practice.