

Building Robots to Learn Design and Engineering

Fred G. Martin*

September 1, 1992

Each year for the past three years, over one hundred and fifty undergraduates at the Massachusetts Institute of Technology have participated in the “LEGO Robot Design Competition,” an experimental class based on the central activity of building a fully functional autonomous robot. This is an unprecedented number for a purely voluntary course, held over the university’s month-long break between semesters. The popular course immerses students in an intensive, hands-on design experience.

Working in teams of two to four, students are introduced to key ideas in engineering and technology: electronic hardware, software design, mechanical design, control theory, and systems integration. More important, the course gives students the opportunity to design, to take their own ideas from initial conception to implementation, debugging, and application. In addition, the final contest pushes students to confront real-world engineering issues of performance, reliability, and resource allocation.

The pedagogical approach taken by the Robot Design class has roots in the constructionist theories of learning developed by Seymour Papert[4]. According to constructionism, the acquisition of knowledge,

*The Media Laboratory at the Massachusetts Institute of Technology, 20 Ames Street Room E15-301, Cambridge, MA 02139. Phone (617) 253-7143. E-mail: fredm@media.mit.edu. This paper is published in the proceedings of the 1992 “Frontiers in Education” conference, sponsored by the I.E.E.E. and the American Society for Engineering Education.

skills, and abilities is an active process of creation engaged in by the learner. This process can be catalyzed when the learner is building something in the world in addition to building knowledge inside his or her own mind.

Based on this theory of learning, the Robot Design class offers a compelling model of a classroom and workshop experience for university-level engineering students. By recognizing the central role that design can have in the learning process, the class inspires high-quality learning, genuine interest, and confidence in students at a variety of stages in their academic careers. Students learn not only about the technical issues mentioned, but also about teamwork and project management, and they experience the engineer's satisfaction and exhilaration of bringing one's ideas into reality.

A Student's Story

The following introductory scenario could have been told by any of the students who have participated in the course:

Before I left M.I.T. for the December term break, my dormmates and I registered as participants in The LEGO Robot Design Competition. Just after returning to campus, at the course's introductory meeting, my team got our robot-building kit.

Later that evening, we dove into the kit, filled with electronic parts, brightly-colored LEGO bricks, beams, gears, and wheels, a battery, and blank printed circuit boards. This was better than any gift I received from my family over the holidays.

We stayed awake all that night, too excited to sleep. We completed the assembly instructions for the main circuit board. The next day, we went to the supervised lab and tested the board. It turns out we made a few wiring errors, but they were easily fixed.

In following weeks, I was completely involved in the design and construction of my robot: building sensors, prototyping designs in LEGO bricks, wiring motors, and writing computer programs to bring it all together. My team and I had heated discussions about the best approach to win the contest.

After several more all-night sessions, we had a robot ready for the eve of the contest. Hundreds of spectators—fellow students, professors, and community parents and children—were present to see the contest; forty robots competed!

My robot made a decent showing, but it didn't win the contest. Still, I feel proud. Not only did I accomplish the task, but I learned how to be an engineer.

The LEGO Robot Design Competition (M.I.T. course number “6.270”) began in 1987 as a student-organized programming contest, inspired by M.I.T.'s own “Introduction to Design” course (2.70) for mechanical engineering students, which was developed by Professor Woodie Flowers[2].

In Flowers' design course, now an institution at M.I.T. for over twenty years, students are each given a kit of identical parts at the beginning of the term: scrap wood, metal, plastic, and motors, and other valuables donated by sponsoring companies. In addition to this kit of assorted “stuff,” students are given the specifications of a competitive task. Their goal is to build a remote-controlled machine that will solve that task faster and better than the other students' machines. The contest specification, which changes every year, might include such tasks as collecting and/or depositing tennis balls, empty soda cans, or a collection of similar objects.

I joined “6.270,” a software-based robot design activity and contest in 1989, and we began a process of transforming it to a month-long course founded on hardware- and software-rich robotic technology. The course is now based on a versatile robot-building kit:

Microprocessor Controller Board. Each student team receives a three-by-five inch, custom-designed microcontroller board that serves as the “brain” of their robot. The board features outputs for up to six DC motors, inputs for over thirty analog and digital sensors, an LCD screen for displaying run-time messages, 32 kilobytes of battery-protected memory, and an eight-bit Motorola CPU.¹

¹The CPU is the Motorola 6811, an eight-bit microprocessor with many features designed for embedded control applications, including a built-in UART, an analog-to-digital converter with an 8-input multiplexer, and a versatile timer and counter subsystem. These parts have been donated to the course by Motorola, Inc.

Interactive, Multi-tasking “C” Programming Environment. The microcontroller board is programmed using a custom-designed “C” language cross-compiler. The compiler is unusual in that it provides an interactive interface: rather than being based on the typical “compile-download-run-debug” program development cycle, the compiler allows the user to interactively execute full expressions at a command-line prompt. This system combines the interface of an interpreted language with the performance of a compiled one.

In addition, the language system supports preemptive multitasking, allowing about eight or ten “C” procedures to run simultaneously.

Robotic Sensors. Students are provided with electronic components and assembly diagrams for constructing a variety of common and useful robotic sensors, including touch switches, resistive stress sensors, photocell light sensors, motor force feedback, shaft encoders, and linear position sensors.

An infrared detection system mandates that robots broadcast encoded infrared light from a supplied infrared beacon; robots may optionally use infrared sensors to detect the angular position of the opponent robots.

LEGO Technics™ System. The mechanical aspect of the robot construction is accomplished with the sophisticated building system developed by LEGO Systems, Inc., trademarked “LEGO Technics.” Far surpassing, while remaining compatible with, the ubiquitous LEGO building brick, the LEGO Technics system includes beams, gears, axles, wheels, chain link, universal joints, and a host of other interlocking mechanical components. These materials have a tremendous degree of versatility, and lend themselves well to incremental and exploratory design work.

Motors. The robot kit includes six DC motors and one airplane servo motor.

Battery System. Robots are powered by a six-volt rechargeable lead-acid battery for motor power and four AA alkaline batteries for

microprocessor power.

In addition to these materials, students each receive a course note manual, which thoroughly explains all the practical aspects of using the technology[3]. Included in the course notes are sections on: how to solder, step-by-step assembly directions for the microprocessor board, construction diagrams for the sensors, a manual for the C programming environment, an explanation of the design of the microprocessor board, and other practical information that will assist them in their robot design project.

Course Organization

The course takes place over M.I.T.'s *Independent Activities Period* (IAP), an unusual one-month time slot between the fall and spring semesters. During IAP, various members of the M.I.T. community, including students, faculty, and staff, offer each other an assortment of lectures, workshops, and other educational endeavors. IAP began as an experimental educational experience in the 1960's; attendance to any of its programs is purely optional (though many of the instructors require a commitment from their participants).

The LEGO Robot Design Competition began and continues as a student-organized IAP activity. Students who take the course may opt to receive elective pass/fail credit.

Most students sign up for the course after forming themselves into teams of two to four members; a few students sign up as individuals and are randomly formed into teams of three. The 1992 version of the course was oversubscribed by more than a factor of two: about 110 teams (341 individuals) signed up for the fifty teams slots the course offers. (A lottery was held to choose the participants.)

In the first meeting of the course students are given the robot kit, and the rules of the robot competition, which will be held one month later, are explained. They are presented with the suggested schedule for their design project:

Week One. Constructing, testing, and debugging the microprocessor

controller board, wiring some sensors, constructing motor assemblies.

Week Two. Preliminary robot designs, exploratory programming using motors and sensors.

Week Three. Finalizing and implementing robot design.

Week Four. Debugging and perfecting design.

Students are strongly encouraged to complete the construction and testing of the microprocessor board within the first week, but it is up to them to organize their own time. There are no formal requirements to have a particular “milestone” accomplished at a particular date. In effect, the only “required” meetings of the course are the first one, in which students receive their kit, and the final two, which are the first and final rounds of the contest itself.

Scheduled course meetings consist of lectures (one per week), recitations (three sections of which a student attends one per week), and supervised laboratory time (afternoons five days per week). Attendance to any of these schedules sessions is optional; most students choose to attend the lectures and recitations, while a fair many students opt to work away from the supervised laboratory as a matter of convenience.

The three course lectures consist of the following:

Introduction. This is the first meeting of the course which provides an overview of the course and schedule, distribution of robot kits, and a discussion of contest specification and rules.

The Art of Building with LEGO. This is a discussion of basic properties and design strategies for building with the LEGO Technic system, including ideas of gear reduction, structural rigidity, and special tricks for using the material in non-obvious ways.

Robust Robot Programming. This lecture presents some ideas for how to develop a control program that allows the robot to recover from unanticipated situations, and in general to perform reliably.

The recitations provide a more interactive format for a classroom setting with the students, covering topics such as: introduction to the C programming language, robotic sensors, and the architecture of the microprocessor control board.

It has been our experience in administering the course over the past four years that the most valuable learning for the students occurs outside of the more traditional lecture and recitation sections: in the lab and in the students' own living groups, when they are working on their projects.

Results

Students who participate in the Robot Design Competition gain practical "how-to" engineering knowledge, learn technical points that complement their more formal education, and get the chance to perform design work, so often lacking in the undergraduate curriculum.

The materials of the robot design kit provide ample opportunity to learn about a number of important concepts, from several academic fields:

Mechanical Engineering. Gear reduction, geartrain design, structural stability, and balance.

Electronic Engineering. Sensors, practical applications of Ohm's Law, power electronics, microprocessor interfacing, board level design.

Software Design. Programming in a procedural language, real-time control, use of multi-tasking software.

Control and Systems Integration. Integration of hardware and software, feedback and stability, system robustness and reliability.

Often these lessons are learned through mistakes and small failures as any number of mishaps can and will occur when people work with hardware. Students may short out lead-acid batteries, and learn that huge amounts of current flow create a fire hazard. Some students

design a system that is so complex that they cannot debug it in time for the contest. These lessons, learned first-hand and sometimes under adversity, are often more memorable than sentences in a textbook.

Analysis and Evaluation

The technology in use in the course has greatly evolved over the four years. As both a designer of the course technology and a teacher of the course itself, I have had the opportunity to reflect on the effects of different materials on the students' work. The following observations come from my analysis of both the development of our course materials and the experiences of the students in the class.

Design environments and materials should encourage playful design and creative exploration.

Part of the success of the course is due to the ease of use of the robot-building kit. Observation has determined that the majority of students, particularly freshmen and sophomores, have not had a great deal of hands-on experience with design and engineering—most of their knowledge is “textbook” knowledge. (Of course, there are students who have a practical background from their personal interests, but these are the exception rather than the rule.) As such, most students are likely to be hesitant and somewhat fearful when it comes to engaging in a practical, hands-on engineering experience—unless the materials are so friendly and easy to use that the students are not deterred.

An earlier year of the course used a microprocessor board that needed to be programmed in assembly language. With this system, there were many obstacles that needed to be overcome before a student could be successful in writing even the most basic program. For example, to input a sensor value and use it to control a motor, many obscure details about assembly language programming would need to be included. Certainly, there was much to be learned about the hardware of a CPU, but the obstacles made the system unfriendly to use, and many students were stymied.

Further, if a system actively encourages playful exploration and iterative design, then a whole new avenue of design work becomes

possible. For example, the familiar and friendly LEGO Technics kit is adored by the students. Although it is complex and rich, students are anxious to explore and learn by playing with the material.

In contrast, much academic design teaching is predicated on a “top-down” model of the design process, in which needs are analyzed, constraints are determined, and an optimization process used to guide the solution (see for example [1]). While it is certainly true that much professional design is based on these practices, there is growing evidence that this is not the preferred working style for many designers. For example, Turkle and Papert [7] revive Levi-Strauss’ term “bricolage” to describe a “bottom-up” method of designing software:

While hierarchy and abstraction are valued by the structured programmers’ planner’s heuristic, bricoleur programmers prefer negotiation and rearrangement of their materials.

Donald Schön, the design theorist, studies the work of professional designers from a variety of fields, including engineering. Schön characterizes design as “a conversation with the materials of the situation”:[6]

There are more variables—kinds of possible moves, norms, and interrelationships of these—than can be represented in a finite model. Because of this complexity, the designer’s moves tend, happily or unhappily, to produce consequences other than those intended. When this happens, the designer may take account of the unintended changes he has made in the situation by forming new appreciations and understanding and making new moves. He shapes the situation in accordance with his initial appreciation of it, the situation “talks back,” and he responds to the situation’s back-talk.²

In the Robot Design course, students are free to choose the style of design which either suits them best or with which they are most familiar. While some students do take a “top down” approach towards their projects, enough do not so that we as educators should take notice. Most students’ projects evolve iteratively and in a piecemeal fashion:

² *The Reflective Practitioner*, pp. 79.

they build on top of their previous efforts, sometimes pausing to re-design a previously working mechanism or integrate several previously separate ones. I believe that for these students, the ideas and concepts they are working with are quite unfamiliar, and by giving them the opportunity to “mess around” with the ideas, they are getting a chance to become truly comfortable with the concepts without the pressure of producing a finished work at first.

Materials should provide a useful level of abstraction, but not limit the students’ explorations.

The robot design kit developed for the students’ use is “user-friendly” because it provides a clean level of abstraction. Students do not need to know how a light sensor functions in order to build one and use it effectively. They do not need to know the low-level details of the microprocessor controller in order to write C programs for it. They do not need to know how the servo motor control is implemented in order to use the motor.

These design decisions were conscious as we were developing the students’ kit. It was more important that the students have an opportunity to *use* these materials, and get excited about the creative potential of these technologies, than it was necessarily that they understand the low-level details of how the various technologies function.

Yet, at the same time, it was important not to limit the students’ explorations. If a student wanted to learn the detail of the interface for a sensor, so that she or he could develop a better sensor, an opportunity should be available. Similarly for any number of topics which were in fact explored by various interested students: new motor control hardware, alternative programming languages, additional digital hardware for more sensors, etc.

In this way, the course materials and also the course philosophy accomodates a wide range of students and their interests. If some students want to concentrate in some areas while ignoring some others, they are encouraged and granted the opportunity to do so.

Students know the difference between authentic learning situations and contrived pedagogical methods.

One of the great strengths of the course is the contest, which is also its culmination. Students take a great deal of pride in readying their robots for this event; many of them literally stay up for two or three consecutive nights in attempts to get their robots in the best possible operating condition.

While a competitive situation is not without its drawbacks, I believe that the contest lends an authenticity to the design work that does not usually occur in academic learning situations. Students are not working for a grade; the course is ungraded, taken for pass-fail credit, if for credit at all. Students are responding to their own satisfaction in engineering, and they are working for the genuine admiration of their peers and teachers: everyone at an engineering school “loves a good hack.”

The contest also forces students to confront the limitations of their design. Were it not for the performance situation, few students would realize how unreliable their designs actually are. Most students, while testing their machines, do not realize that one successful performance, which might be contingent on the correct operation of a dozen different actions, does not imply that those dozen actions can be relied upon. Yet many do realize this after their machines have performed in the actual contest, facing several different types of opposing robots.

Closing

The M.I.T. LEGO Robot Design Competition is a work-in-progress. Still organized as a student activity, it enjoys a certain freedom from the constraints of a filling a spot in a formal university curriculum. Yet its success from this position may have lessons for those who design and choose the required courses of an undergraduate engineering degree.

Several students have actually approached me to express gratitude after taking the course, commenting that this was the first experience they had at M.I.T. that gave them the chance to express their own ideas. Undergraduate education should provide this type of activity by design rather than by chance.

Acknowledgments

Wanda M. Gleason provided helpful and timely feedback on the writing of this draft.

The collaboration with two other individuals in running the Robot Design Competition was of great value to both myself and the project. Randy Sargent and I worked as an inseparable team when we were developing the technology now used in the course. Pankaj Oberoi provided a steadfast organizational and inspirational role over the past few years.

Several corporations both large and small have provided much needed support, financial and otherwise, to our project. Microsoft Corp. and Motorola, Inc. were our early and on-going main sponsors, joined by Polaroid Corp., LEGO Systems Inc., Methode Inc., Gates Energy Products Inc., 3M Inc., and Abrams-Gentile Entertainment, Inc.

I am also grateful to the generous support of M.I.T.'s Electrical Engineering and Computer Science Department, which seemed to "recognize a good thing" early on. In particular I would like to thank Professor Leonard Gould for his support.

The unflagging support of my advisor, Professor Edith Ackermann, and my research group, the Epistemology and Learning Group, headed by Professor Seymour Papert, as well as our laboratory, the M.I.T. Media Laboratory, has been essential.

For More Information

The technology that has been developed for the M.I.T. LEGO Robot Design Competition—including the microprocessor control board, the Interactive C compiler software, and the 250-page *6.270 Robot Builder's Guide*—is available for use with no licensing fee.

At the time of this writing, the sole method of distribution is via anonymous FTP over the Internet network. A server machine named `cherupakha.media.mit.edu` (Internet address 18.85.0.47) stores all of this material, in the directory `pub/6270`. (If your computer network does not allow anonymous FTP access, you may be able to use the FTPMAIL facility supported as a public service by Digital Equipment

Corporation. Send an e-mail note containing the single word “help” to the address `ftpmail@decwrl.dec.com` for instructions.)

References

- [1] Morris Asimow. *Introduction to Design*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1962.
- [2] Woodie C. Flowers. On engineering students’ creativity and academia. *ASEE Annual Conference Proceedings*, 1987.
- [3] Fred G. Martin. *The 6.270 Robot Builder’s Guide*. 1992. Course notes for the M.I.T. LEGO Robot Design Competition, Epistemology and Learning Group at the M.I.T. Media Laboratory.
- [4] Seymour Papert. Constructionism: A new opportunity for elementary science education. Proposal to the National Science Foundation. MIT Media Laboratory, 1986.
- [5] Henry Petroski. *To Engineer is Human: The Role of Failure in Successful Design*. St. Martin’s Press, New York, 1982–1985.
- [6] Donald A. Schön. *The Reflective Practitioner: How Professionals Think in Action*. Basic Books, Inc., 1982.
- [7] Sherry Turkle and Seymour Papert. Epistemological pluralism: Styles and voices within the computer culture. *Signs: Journal of Women in Culture and Society*, 16(1), 1990.