A Robot Laboratory for Teaching Artificial Intelligence*

Deepak Kumar
Department of Mathematics
Bryn Mawr College
Bryn Mawr, PA 19010
(610) 526-7485
dkumar@cc.brynmawr.edu

Lisa Meeden
Computer Science Program
Swarthmore College
Swarthmore, PA 19081
(610) 328-8618
meeden@cs.swarthmore.edu

1 Introduction

There is a growing consensus among computer science faculty that it is quite difficult to teach the introductory course on Artificial Intelligence well [4, 6]. In part this is because AI lacks a unified methodology, overlaps with many other disciplines, and involves a wide range of skills from very applied to quite formal. In the funded project[1] described here we have addressed these problems by

- Offering a unifying theme that draws together the disparate topics of AI;
- Focusing the course syllabus on the role AI plays in the core computer science curriculum; and
- Motivating the students to learn by using concrete, hands-on laboratory exercises.

Our approach is to conceive of topics in AI as robotics tasks. In the laboratory, students build their own robots and program them to accomplish the tasks. By constructing a physical entity in conjunction with the code to control it, students have a unique opportunity to directly tackle many central issues of computer science including the interaction between hardware and software, space complexity in terms of the memory limitations of the robot's controller, and time complexity in terms of the speed of the robot's action decisions. More importantly, the robot theme provides a strong incentive towards learning because students want to see their inventions succeed.

This robot-centered approach is an extension of the agent-centered approach adopted by Russell and Norvig in their recent text book [11]. Taking the agent perspective, the problem of AI is seen as describing and building agents that receive perceptions as input and then output appropriate actions based on them. As a result the study of AI centers around how best to implement this mapping from perceptions to actions. The robot perspective takes this approach one step further; rather than studying software agents in a simulated environment, we embed physical agents in the real world. This adds a dimension of complexity as well as excitement to the AI course. The complexity has to do with additional demands of learning robot building techniques but can be overcome by the introduction of kits that are easy to assemble. Additionally, they are lightweight, inexpensive to maintain, programmable through the standard interfaces provided on most computers, and yet, offer sufficient extensibility to create and experiment with a wide range of agent behaviors. At the same time, using robots also leads the students to an important conclusion about scalability: the real world is very different from a simulated world, which has been a long standing criticism of many well-known AI techniques.

We proposed a plan to develop identical robot building laboratories at both Bryn Mawr and Swarthmore Colleges that would allow us to integrate the construction of robots into our introductory AI courses. Furthermore, we hoped that these laboratories would encourage our undergraduate students to pursue honors theses and research projects dealing with the building of physical agents.

2 Incorporating a Robot Laboratory into the AI Course

In designing our syllabus for this new version of introductory AI, we tried to strike a balance among three goals: focusing on the robot theme, including only the

*Proceedings of the Twenty-ninth SIGCSE Technical Symposium on Computer Science Education, ACM SIGCSE Bulletin, D. Joyce (editor), Volume 30, Number 1, Pages 341–344, ACM Press, March 1998.

[1]This work was sponsored in part by grant DUE-9651472 from the National Science Foundation's ILI/IP program, grants from Bryn Mawr College, Swarthmore College, The Mellon Foundation, and The Howard Hughes Medical Institute.

most important ideas from traditional AI, and emphasizing AI's connection to core computer science[2].

The first goal is in reaction to the misconception that AI is a collection of unrelated subfields which arises when an AI course is presented from the historical perspective (as is adopted by most AI text books [2]). We, as educators, must emphasize the field's common problems and standard techniques for attacking them. The robot theme enables various components of intelligence to be gathered together toward a unified goal: building a rational, competent physical robot.

The second goal addresses an important pitfall in the design of an AI course syllabus: the need to identify and eliminate concepts traditionally covered in older offerings that are either no longer relevant or are incorporated in other core computer science courses. Two such examples are: the coverage of basic graph searching techniques; and the teaching of Lisp/Prolog. The ACM-IEEE curriculum recommends the coverage of basic graph searching techniques in the second introductory computer science course as well as in one of the discrete mathematics courses. The teaching of Lisp/Prolog, traditionally relegated to the introductory AI course is now covered in the core computer science curriculum in either the introductory course itself, or as a part of the programming language concepts course, or in a separate course on programming paradigms. At Swarthmore College, students learn Scheme (a derivative of Lisp) in the introduction to programming course, and use it again in the programming language course. Bryn Mawr College offers a separate, sophomore-level, course titled, Programming Paradigms, that introduces Lisp and Prolog and serves as a pre-requisite for the AI course [7]. Consequently, our AI courses do not need to dwell on teaching the Lisp language, but can focus on the fundamental questions of AI itself.

The third goal identifies the need for placing AI in the context of computer science, for example as a study of advanced algorithms. In this perspective, the algorithms presented in AI contribute to enhancing the skill and overall interest in the field of computer science. For instance, one can compare compilation techniques with algorithms for natural language parsing, and one can relate logical reasoning to pattern matching and search.

3 Outfitting a Robot Laboratory

The robot building laboratory is essentially a low cost extension of an existing computing laboratory. Each computer workstation is augmented with additional external hardware and materials and can serve teams of 2-3 students in a robot exercise. The following materials are needed for each robot station:

[2]For another perspective on using robots in an AI course see [12]

- A computer (PC/MAC/UNIX) with an available serial port
- A robot controller board
- Control software
- Robot building materials: LEGO Technic Resource sets, sensors, motors, etc.

The key hardware component, besides the computer, is the robot controller board. We are using a controller board called the Handy Board [8]. It is a third generation controller board, preceded earlier by the Miniboard [9], and the 6.270 board [3].

The Handy Board is based on a Motorola 6811 microcontroller that has a built in 8-bit A/D converter. It is powered by rechargeable batteries. The complete unit is a single, solid-state package measuring approximately 4x3x1 inches and weighs less than 1lb. The board provides a serial interface for connecting to the computer, two programmable input switches, 7 analog input ports, and 8 digital output ports. Additionally, the board has an infrared receiver capable of accepting signals from a standard television remote, a piezoelectric beeper, and a 2-line LCD display. The board also includes 4 ports for controlling analog output devices that are most commonly used for controlling DC motors. A variety of off-the-shelf sensors can be easily interfaced via the input ports.

Control programs for the Handy Board are written in Interactive C (IC). IC is a subset of the C programming language augmented with library routines for access to the interface components of the board (switches, input and output ports) as well as for multi-tasking. Interactive C software runs on several computer platforms (MAC/PC/UNIX). IC programs are written by users on the workstation and then downloaded for execution on the Handyboard via the serial interface. Currently, there are at least three vendors that sell Handy Boards and IC. The total cost of these two units is approximately $300.

Several materials can be used for building the bodies of robots. In our laboratory, we use LEGO Technic materials. It is also possible to build mobile robot bodies by retrofitting toy cars. Several hobby stores also sell ready-made bodies for small mobile robots. Figure 1 shows a LEGO-based robot with the Handy Board mounted on it.

4 Robot Exercises

The course has weekly, 2-hour, laboratory sessions. Students form groups of 2-3 to work jointly on their robot exercises. Each group is issued a robot kit that can be carried out of the laboratory in a small activity box.

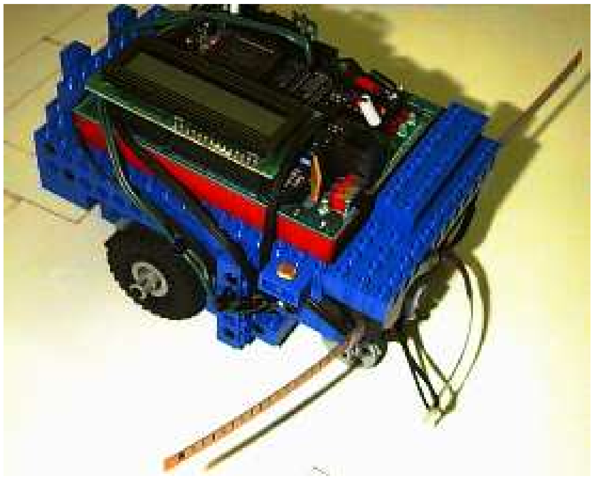

Figure 1: A LEGO-based robot with the Handyboard

Laboratory sessions in the first half of the semester are conducted as closed laboratories. The objective of these sessions is to introduce robot building materials and to familiarize students with standard techniques of interfacing sensors and writing simple robot control programs. Each laboratory session has structured exercises. Students start the exercises in the laboratory and spend the rest of the week completing it. Robots resulting from previous week's exercises are demonstrated at the beginning of the next week's laboratory session. A laboratory manual containing these exercises has been prepared and is available via the world-wide web [5].

One topic that we were able to explore hands-on in the robot laboratory was behavior-based programming. Using this paradigm robot programs are constructed as a hierarchy of behavior modules that execute concurrently in a multi-tasking environment. This enables a robot builder to create robots that exhibit simple behaviors to which more sophisticated behaviors can be added simply by coding higher level behavior modules. This way of organizing a hierarchy of behavior modules is also termed *subsumption architecture* [1]. This proved to be a unique and concrete way of introducing multi-tasking, and issues relating to real-time control in the computer science curriculum.

Laboratories in the second half of the semester are conducted as open laboratory sessions. Rather than giving structured weekly exercises, students are asked to design their own robot tasks, and robots to fulfill them. The only requirement for these half-term projects is that they should employ some of the concepts and techniques of AI discussed in the lectures (or in the texts). At the end of the semester, each group gives a presentation on their half-term projects.

The laboratory exercises address the problem of building an intelligent robot from scratch. There are two main steps involved: developing the hardware and then writing the software to control that hardware. Typically, the tasks only provide an abstract description of the desired mapping from perception to action. Much of the detailed specifications about how to behave in a particular situation is not given and may not be known. For example, a typical robotics task is to navigate from a starting point to a goal point. How will the robot recognize that it has arrived at the goal location? What obstacles can the robot expect to encounter along the way? If a map is available, how do positions on the map correspond to the robot's sensor readings? Due to the open ended nature of robotics problems, a lengthy process of trial and error is often necessary to answer these types of questions clearly enough to develop a working algorithm. Students must repeatedly run their programs on the robot and watch how well it performs. They must analyze the failures and determine how they emanate from the program. They must then modify the programs in order to correct the failures or improve the performance. During this debugging period, students learn a great deal about the interaction between the robot's sensors and its physical environment, and in turn must translate this knowledge back into their programs.

Another key area that is incorporated into the laboratory is the idea of working in teams. By working on the same project together students are able to accomplish more than what they could alone. Though not everyone in a team has the same level of expertise in programming, engineering, and structural design, team organization provides an opportunity to learn from each other.

5 Robots and the CS Curriculum

A robot building laboratory intensifies the role that an AI course can play in the overall computer science curriculum in several important ways. Typical undergraduate computer science curricula focus mostly on the development and analysis of algorithms. Thus, indirectly, the emphasis is on software development. This creates a void in that the students may never experience the underlying hardware of the computers they utilize. Students are required to take a core course on computer organization and perhaps an elective on computer architecture. However, even in these courses, there is minimal exposure to actual physical hardware components. Construction of physical agents intrinsically involves the handling of controller boards (microcomputers in themselves), interfacing of sensors to these boards, and, more importantly, dealing with the physical connection between a computer and the controller board (via a serial port). Developing software to control the behavior of physical robots also involves working under constrained resources (especially with respect to speed and memory) imposed by the robot control board. This provides a direct exposure to the time as well as the space complexity of algorithms. Yet another link that can be emphasized via the laboratory exercises is that between the algorithms embodied in the agents and their equivalence as a whole to an automaton.

We have also used the robot building laboratory as a basis for teaching introductory computer science [10]. A new course titled, *Building Intelligent Robots* was offered at Swarthmore College to serve as a broad introduction to computer science as well as artificial intelligence. Since this course had no prerequisites, students came from a wide variety of disciplines including economics, engineering, literature, mathematics, psychology, and religion. This course stressed both application and theory by dividing class time equally between laboratory sessions and discussion sessions. Robots fascinate the typical undergraduate student, and we should use this interest to draw students into the computer science curriculum. Building a robot from start to finish is an intensely engrossing experience that motivates the students to learn about many of the less glamorous theoretical aspects of computer science.

6 Dissemination

We have designed our laboratories so that our experiences can be easily duplicated at other institutions. We have restricted ourselves to the use of standard equipment, interfaces, and software that is freely available. The key outcome of this project is a laboratory manual which is specifically tailored towards the use of robotic agents in the context of an AI course at an undergraduate level. This is not unlike similar laboratory manuals now available for several introductory computer science courses. While the need for development of closed laboratory materials for introductory courses has been identified, to our knowledge, similar material is not available for upper-level computer science courses. The resulting laboratory manual and any associated materials are available via the web.

7 Summary

In this paper, we have described a project that involved the creation of two identical robot building laboratories at two undergraduate institutions. The laboratory has been used for teaching the undergraduate AI course. Laboratory materials have been made available along with detailed information on how to create such a laboratory at other institutions. The laboratory helps introduce a central theme in the teaching of AI, and, at the same time, strengthens the role AI plays in the core computer science curriculum.

References

[1] Rodney A. Brooks. A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, March 1986.

[2] Eugene Charniak and Drew McDermott. *An Introduction to Artificial Intelligence*. Addison Wesley, 1985.

[3] Matt Domsch. MIT 6.270 LEGO Robot Design Competition. World Wide Web, URL is http:// www.mit.edu / courses / 6.270 / home.html.

[4] Marti A. Hearst. Preface: Improving Instruction of Introductory Artificial Intelligence. In *Working Notes of the 1994 AAAI Fall Symposium on Improving the Instruction of Introductory Artificial Intelligence*, pages 1–4. AAAI Technical report, November 1994.

[5] Deepak Kumar. Introductory AI Course Syllabus. World Wide Web, URL is http: // blackcat.brynmawr.edu:80 / CS / CS372 / Materials /.

[6] Deepak Kumar and Marti A. Hearst, editors. *ACM SIGART Bulletin: Special Issue on Artificial Intelligence Education*, volume 6. ACM Press, April 1995.

[7] Deepak Kumar and Richard Wyatt. Undergraduate AI and its Non-imperative Prerequisite. *ACM SIGART Bulletin: Special Issue on Artificial Intelligence Education*, 6(2):11–13, April 1995.

[8] Fred Martin. The handy board. World Wide Web, URL is http:// lcs.www.media.mit.edu / groups / el / Projects / handy-board /.

[9] Fred Martin. Mini Board 2.0 Technical Reference. MIT Media Lab, Cambridge MA, 1994.

[10] Lisa Meeden. Using Robots As Introduction to Computer Science. In John H. Stewman, editor, *Proceedings of the Ninth Florida Artificial Intelligence Research Symposium (FLAIRS)*, pages 473–477. Florida AI Research Society, 1996.

[11] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Englewood Cliffs, NJ, 1995.

[12] Carl Turner, Kenneth Ford, Steve Dobbs, and Niranjan Suri. Robots in the classroom. In John H. Stewman, editor, *Proceedings of the Ninth Florida Artificial Intelligence Research Symposium (FLAIRS)*, pages 497–500. Florida AI Research Society, 1996.