

# Teaching Bottom-Up AI From the Top Down

**Christopher Welty, Kenneth Livingston, Calder Martin,  
Julie Hamilton, and Christopher Rugger**

Cognitive Science Program  
Vassar College  
Poughkeepsie, NY 12604-0462  
{*weltyc, livingst, camartin, juhamilton, chrugger*}@vassar.edu

From: Proceedings of the Eleventh International FLAIRS Conference. Copyright © 1998, AAAI (www.aaai.org). All rights reserved.

**Abstract.** For many reasons, it is desirable to use robots in courses such as introductory computer science, artificial intelligence, and cognitive science, yet the knowledge normally required by students to make effective use of these tools is often prohibitive in such courses with well established curricula. We have developed a user interface that allows students with no prior experience or training in robotics to experiment with behavior networks in real robots, and then brings them down through the software and then the hardware involved. The interface is still in the early stages of development, and has been tested somewhat in a Cognitive Science course, but more widespread use is expected.

**Subject Areas:** AI Education, Cognitive Science Education, Robotics.

## 1 Introduction

Robots are very useful tools in education. In computer science, they can be used to provide immediate feedback on performance, and force students to understand resource limitations and efficiency, among other things [Meeden, 1996]. In Artificial Intelligence, they can be used to demonstrate dynamic planning, genetic algorithms, and to compare various control strategies such as the classic bottom-up vs. top-down approaches [Turner, et al., 1996].

In Cognitive Science, robots are also useful to demonstrate specific instantiations of various sensor-motor behaviors, and explore the various architectures for perception and action. Small robots can be used to isolate very simple parts of what may normally be a larger system, and study them.

In addition, and perhaps most importantly, students find robots fascinating, and get a great sense of accomplishment even after building or programming the simplest systems and watching them work. Using robots is an accepted method for intensifying student participation [Kumar and Meeden, 1997], not to mention press coverage.

For these reasons and others, using robots in the classroom in general Computer Science, AI, and Cognitive Science, is very desirable.

There are, however, numerous obstacles to bringing robots into the classroom. They are expensive, for one, and require a lot of maintenance. The skills to use them are fairly difficult to obtain, and thus the resources required to keep them working are hard to come by. Robots, and robotics in general, can actually be quite intimidating to students outside of Computer Science (specifically, Cognitive Science), and this problem, at least through our experiences, is getting worse, not better. In fact, these are typically students who would benefit the most from exposure to this technol-

ogy.

The primary obstacle to using robots in the classroom is time. Few schools with established Computer Science or Cognitive Science undergraduate degrees really have room in the curriculum, or the staffing, to offer a course in robotics. In order to use robots to learn the advanced concepts mentioned above, students must first spend inordinate amounts of time learning the very specific, and in many cases tedious, aspects of circuits and electronics, and very low-level programming. These are topics not normally taught anymore. In existing courses, such as an undergraduate AI course, CS I, or a Cognitive Science class like Perception and Action, the syllabus is already crammed full of material. There simply is not room for robotics because even the simplest activities require too much background and preparation.

These problems are only enhanced in a liberal arts setting, where very technical courses are rare, and in Cognitive Science, where there are many students who fear technology. Introducing robots in such an environment typically yields to frustration on the part of the students, because in the early steps (soldering, configuring sensors, testing components, etc.) they have no clear vision of how these low level activities relate to what they signed on to learn. This frustration, and also time commitment, turns students away.

## **2 Our Approach**

This project started with the desire to introduce rug-warrior style robots [Jones and Flynn, 1993] into the Perception and Action course, which is part of the core curriculum in Vassar's Cognitive Science Program. The course already delved, at a very high level, into behavior networks [Brooks, 1986], and spent a lot of time developing models of P&A, yet attempts to incorporate robots met with

repeated failures, due to the problems outlined above. The students, with a few exceptions, found the material tedious, uninspiring, or intimidating.

One of these exceptional students, who was, on the contrary, inspired by the idea of small robots and bottom-up approaches to AI, decided that the failure of the course for his peers was due to the way the material was being presented. For his senior thesis, he developed a prototype of an interface that allowed students to configure the behavior networks on the screen, then generate code (in Interactive C) that implemented the network and download it onto a robot.

### **2.1 Taking it from the top**

The basic revelation of this senior thesis was that bottom-up approaches, such as those manifested by small robots, do not need to be presented from the bottom up, i.e. starting with circuit boards and infra-red detectors, but can be presented from the top down, i.e. starting with the notion of perceptions and their associated actions in simple reactive systems, and demonstrating this behavior in working robots.

This is, although a simple approach, quite a departure from the way robots are traditionally presented to students. It took us quite a while to realize that most of the obstacles to using robots in the classroom actually stem from the traditional approach. Much of what we want use robots to teach, however, is not preconditioned on understanding the circuits and electronics of the robot itself. Once students are drawn into the subject matter of robots, as time permits we slowly move them down towards the lower-level understanding of the robots themselves. The students are drawn further and further in by the increased control they get over an increasing range of possibilities. This as opposed to the frustration of experience short-circuits because of sloppy soldering.

## 2.2 An Interface

We have developed an interface that consists of three levels, each level deepening the students knowledge of the robots. Our goal is to empower the student to progress further into the subject matter as independently as possible.

The three levels we implemented are: behavior nets, software, and hardware.

At the behavior net level, students manipulate graphs that look like the behavior net suppression hierarchies presented in most texts on the subject. By reordering the nodes in the graph, condition-action pairs are changed in precedence, the higher pairs overriding any below them when their conditions are met (by specific sensor states), and the action (specific actuator states) are carried out. The sensor and actuator states are represented symbolically as the conceptual behaviors they represent, e.g. *cruise*, *seek light*, *follow*, etc.

Level one, the behavior net level, is broken into two parts: the first part involves observing a robot and figuring out the behavior network; the second part involves configuring a behavior net, downloading software that represents that net onto the robot, and observing the robot. These activities are engaging, and the relatively limited set of realistic behavior net configurations immediately draws students into wondering how they can do more.

Level two, the software level, presents programming the robot in Interactive C. We begin here by presenting the code resulting from a behavior net configured in level one, isolating and identifying the procedures that implement each behavior, and allowing the students to make very small changes to the code (such as changing the speed of a motor). This modified code is then downloaded into the robot and the resulting effect is observed. The student is guided through a number of incrementally

more complex programming tasks, culminating in implementing entirely new behaviors and adding them to the set of possibilities presented in level one of the interface.

Level three, the hardware level, presents a tutorial on the hardware used on the robot. This material is presented in a hypertextual tutorial form, with very little interaction with the student, since work at this level is mainly on the circuit board of the robot. Students are provided background on how and why the sensors and actuators work, given tutorials on configuring them, and in particular diagnosing problems. Examples of such tutorials already exist on-line [Martin, 1997].

## 3 Evaluation

A preliminary version of the interface was used in the Fall of 1996, with success. Students were able to configure a very simple set of behaviors and observe the robot in a controlled environment. All the students were able to complete the exercise successfully, and found it enjoyable.

We planned to formally evaluate the interface in the Perception and Action course this fall by splitting the class into two lab sections. Both lab groups are in the same lecture, and so are presented with the same material and reading.

The first lab section is presented with the “guess the behavior net” game that is the first part of level one of the interface. The second lab section plays this game as well, but without using the interface. The scores of the two groups are recorded. The first group then goes on experimenting with the behavior networks themselves, using part two of level one of the interface. They try out different behavior orderings and see how the robot works. The second group proceeds as had been done previously in the course, doing paper problems and considering behavior networks and their effect on a robot, without having a real robot to

experiment with. Finally, the two groups play the “guess the behavior net” game again, and their scores are recorded.

Our expectation here is that the first group will have shown a significantly larger improvement in their scores than the second group.

The purpose of the evaluation is perhaps not immediately obvious. The simple fact is that, without the interface, there is no way to let students experiment with a real robot in the time allotted to this material in the course. Therefore showing that the interface is an improvement is, in that sense, entirely moot. We believe the expected results would show that *using robots in this context enables an improved understanding of the subject material* (in this case, behavior networks). Then, given that we can’t use robots without the interface, the interface becomes a necessary pedagogical tool to realize this improved understanding.

## 4 Results

At the last moment, the evaluation had to be cancelled because the robots broke, and, as mentioned, the course is so packed with material that there was no time to go back to it. The P&A course is only offered in the fall, however we plan to make another attempt in a special (one time only) seminar course on robotics this spring, which we would be able to report on at FLAIRS. In addition, we are scheduled to present this material to high school students this coming March, and plan to run a formal evaluation with that group as well.

Despite the failure of our formal evaluation, our observations of the students using the interface, when all things were working, were quite favorable. Students in particular enjoy playing with different behavior networks and watching the results, and this clearly begs the question for them about what more they can do.

At this point in time, level one of the interface is complete, level two is nearing completion, and for level three we are collecting the tutorial information from existing sources. We have concrete plans to distribute this interface and see it used widely, and we have made arrangements with A.K. Peters Publishers to include our software with the Mobile Robot Kits they sell.

The failure of the empirical evaluation actually brings up another negative point about using robots in the classroom. While the technology for small robots has certainly been stabilizing over the past few years, it is not completely stable. Problems with the robots are frequent, and solving them can be tedious and time-consuming. For students these problems remain a serious obstacle. Spending several hours mulling over C code to find an “=” where an “==” should be is frustrating enough, but spending days dealing with an intermittent problem that turns out to be a faulty connection on the input port is enough to warrant the installation of metal detectors in buildings for the safety of those giving such assignments.

Ensuring redundancy by having many robots and hiring a technician to handle this work is our current solution, and no better one has presented itself.

## 5 Conclusion

We have developed a software interface to mobile robots that presents material from the top-down. Rather than beginning with circuits and electronics, we begin teaching high-level concepts like *behavior networks* [Brooks, 1986]. The interface permits students to manipulate graphical representations of these networks, generate code that implements the resulting behavior precedences, and download the code into a robot.

Once students are drawn in with this simple interaction, they are taken into the program-

ming level, and encouraged to develop their own behaviors in Interactive C. Finally, the hardware level is presented via an illustrated hypertext tutorial.

Student interaction with the interface has produced very positive feedback. The “guess the hierarchy” exercise is very popular and generates a lot of excitement. Configuring the robot graphically and then watching it perform is an excellent chance for students to engage in scientific inquiry.

While the levels are not complete, and no one has made use of the software or hardware levels yet, we are optimistic that these levels will be as successful as those we tested.

## References

[Brooks, 1986] Brooks, Rodney A. *Achieving Artificial Intelligence Through Building Robots*. MIT AI Lab Memo 899, May 1986. Available at <ftp://publications.ai.mit.edu/ai-publications/500-999/AIM-899.ps>.

[Jones and Flynn, 1993] Jones, Joe, and Flynn, Anita. *Mobile Robots: Inspiration to Implementation*. AK Peters. Cambridge, MA. 1993.

[Kumar and Meeden, 1997] Kumar, Deepak, and Meeden, Lisa. Using Robots in an AI Course. *NSF UGAI-97: The 1997 Workshop on Teaching Undergraduate AI*. July, 1997. Available at <http://blackcat.brynmawr.edu/~dkumar/UGAI/Robots.html>.

[Martin, 1997] Martin, Fred. *Building the Lego Bug*. MIT 6270 course notes, May 1997. Available at <http://lcs.www.media.mit.edu/people/fredm/projects/legobug/>.

[Meeden, 1996] Meeden, Lisa. Using Robotics as an Introduction to Computer Science. In *Proceedings of FLAIRS-96: The Florida AI Research Symposium*. May, 1996. Available at <http://www.cs.swarthmore.edu/~meeden/flairs96.html>

[Turner, et al., 1996] Turner, C., Ford, K., Dobbs, S., and Suri, N. Robots in the Classroom. In *Proceedings of FLAIRS-96: The Florida AI Research Symposium*. May, 1996. Available at <http://www.coginst.uwf.edu/~cturner/roboteach.ps>.