Teaching Introductory AI: A Design Stance

Ashok Goel*
AI Group, College of Computing,
Georgia Institute of Technology, Atlanta, Georgia 30332-0280
goel@cc.gatech.edu, (404) 853-9371
August 31, 1994

The Challenge of Teaching AI

Teaching introductory AI is notoriously challenging. My experience in teaching AI suggests that this because of four main reasons:

1. *Fragmentation of AI as a Subject of Study.* AI as a subject of study and research is not only new, it is also fragmented. Beyond a few general goals and core assumptions, there is at present little agreement among AI researchers about even the most basic issues in the field. If a AI teacher attempts to cover various schools of thought in the field, then the discussion runs the risk of becoming vague and disjointed, and the field appears as lacking internal coherence. If, however, the teacher attempts to develop a specific view of AI, for example AI as a cognitive science or engineering, then the discussion runs the danger of leaving out major portions of AI and giving an overly biased and narrow view of the field.
2. *Imprecision of Teaching Goals.* We, as AI teachers, often are not clear about what we want our students to learn. For example, do we want our students to learn some set of concepts or do we want them to learn some set of skills? Clearly the teaching strategies and methods would depend on the precise teaching goals and objectives.
3. *Abstractness of AI Concepts.* Many AI concepts are quite abstract, and, therefore, hard to understand. The concept of 'problem spaces' is a good example of this; 'control strategy' is another example.
4. *Imprecision of Assessment Tools.* It is not clear what, if anything, do our students really learn from introductory AI classes. The standard tools for assessing the learning of students - examinations, project reports, student reviews - are too blunt to precisely indicate what a student has learned. As a result, we (AI teachers) get little useful feedback on our teaching, and, thus, are unable to improve it.

Note that only the first of these four factors is specific to AI. The other three are equally applicable to teaching computer science in general, and probably to all high-level science education.

I am not convinced that there exists a single, simple answer to these issues. Nevertheless, in my introductory AI classes, both undergraduate and graduate, I have found it useful to adopt a design stance towards the teaching of introductory AI. This design stance arises from the view of *AI as a design science* first articulated by Herbert Simon [1981].

Viewing AI as Design Science

In the beautiful little book 'Sciences of the Artificial', Simon characterized AI as a design science. As a design science, AI has the goal of constructing a design theory of intelligence that relates design decisions with intelligent behaviors. Note that the view of AI as a design science is *not* identical to viewing AI as engineering. The difference lies in the role of the computer program. In AI engineering, the computer program represents a solution to a real problem, and can be evaluated by considering how well the program solves the problem. In AI design science, the computer program represents a parameterized model that enables the drawing of a map between the design choices and the behaviors of the model. This is done by tweaking the design parameters of the model and observing its output behaviors. Of course, once the mapping between the design decisions and the output behaviors is known, it becomes possible to engineer a specific program for achieving a given behavior. In addition, the map between design decisions and program behaviors acts both as a generator and a verifier of hypotheses about cognition.

*This work has been supported by the Lilly Foundation through a Teaching Fellowship during 1992-93, and by the Center for Enhancement of Teaching and Learning and the College of Computing at Georgia Tech. It has benefited from discussions at the Cognitive Science Workshop on Teaching Design Skills held at Georgia Tech in March 1992, and discussions of the College of Computing Task Force on Teaching Quality during 1992-93.

Taking a Design Stance Towards Teaching AI

How does the view of AI as a design science help in addressing the four issues I raised earlier? In the design stance I have adopted towards teaching AI, the answers to the first three issues take the following form:

1. *Unification of AI as a Subject of Study.* The view of AI as a design science suggests the choice between teaching AI as a cognitive science or as an engineering is a false one. Since AI design science charts the maps between the internal design decisions and the output behaviors of intelligent systems, it provides both a generator and verifier of hypotheses about cognition, and, in addition, leads to a technology for designing systems for delivering the specific behaviors needed to solve real problems. In this way, it rescues the study of AI from being overly biased and narrow. Further, the theme of design provides a principled basis for studying AI, and gives it a conceptual coherence that saves it from appearing vague and disjointed.
2. *Characterization of the Teaching Goals.* The view of AI as a design science suggests the development of a set of design skills as the goal of studying AI. By design skills, I mean here design competences and abilities, values and strategies.
3. *Contextualization of AI Concepts.* The design theme provides a context for motivating, explaining, illustrating and operationalizing abstract AI concepts.

Note that the design stance does not directly provide an answer to the issue of assessment of learning. I will return to this issue a little later.

Operationalizing the Design Stance

In general, introductory AI classes involve two main kinds of activities. Firstly, the teacher delivers a series of lectures on basic AI facts, concepts, methods and results. Often the lectures explain and illustrate individual concepts and methods through examples but they either do not relate the various concepts and methods into a coherent theme or focus on a narrow subset of concepts as we mentioned earlier. Typically the lectures do not involve the students except as passive listeners. Secondly, the students actively engage in a series of programming projects. Often the projects specify both a problem and a partial solution to the problem (in the form of the algorithm and the data structures) and ask the students to complete the partial solution, code it in some language, and test the code on a given set of data. Typically students work on the projects individually, and the various projects are unrelated to one another and only marginally related to the concepts and methods discussed in the class. From the perspective of learning design skills, the value of these two kinds of activities is quite limited.

The goal of learning design skills, instead, suggests teaching strategies that emphasize design related activities such open-ended design projects, design case studies, construction and evaluation of real designs, exploration and experimentation, articulation and reflection, communication and collaboration, etc [Goel 1993].

Design Case Studies: In the beginning of the course, I trace in some detail the evolution of the design of an AI system. In my undergraduate introductory AI class, I often use the Router system as a case study, where Router is an experimental multistrategy planning and learning system for robot navigation developed by my research group [Goel et al 1992, 1993]. The goal is to illustrate AI in action and ground it in realistic problems. I use Router as a case study to illustrate some of the many kinds of problems that AI seeks to address (e.g., planning and learning in the real world), the kinds of issues that arise in these problems (e.g., competence and performance; representation, organization, use and acquisition of knowledge, strategic control), the kinds of solutions that AI develops (e.g., multistrategy reasoning, plan reuse, performance-driven and experience-based learning). I also use this case study to illustrate the kinds of mistakes we made in designing Router, methods for evaluating AI systems such as Router, and what and how we learned from the evaluation.

Design Projects: Then, I give the students a sequence of interrelated design projects in which each design project builds on preceding ones. The goal is to illustrate the competence and performance differences made by AI concepts and methods. In one undergraduate introductory AI class, in the first project (which also introduced Lisp) the students designed and developed a system that used (a limited form of) depth-first search on a qualitative spatial model of the Georgia Tech campus to form navigation plans. In the second project, they used (a form of) mean-ends analysis on a hierarchically-organized spatial model for the same task of navigation planning. In the third project, they stored the solutions generated by means-ends analysis in a case memory, and formed new navigation plans by retrieving and modifying old cases. The system the students built in the fourth project dynamically and opportunistically selected and instantiated a specific planning strategy at planning time. It also interpreted feedback on the execution of a plan and used the feedback to update the world model; and so on. In this way, through the various design projects, each student designed and developed his/her micro-Router system. Note that the different projects used different methods for the same problem. Note that the complexity of the design increased in the latter projects.

Exploration and Experimentation: I make each design project somewhat open-ended. The goal is to provide students with opportunities for active learning through exploration and experimentation. Instead of fully specifying the problem, I deliberately leave the problem specification somewhat incomplete and ambiguous so that it is open to multiple interpretations. Similarly, I deliberately leave the specification of the AI methods in the project incomplete and ambiguous. I ask the students to explore different problem interpretations and experiment with different AI methods both with in and across families of methods. This exposes the students to use AI concepts and methods for analyzing problems in addition to constructing solutions.

Evaluation and Reflection: I ask the students to evaluate their systems in a number of dimensions including efficiency of processing and scalability, quality of solutions and flexibility, knowledge requirements and generality. In addition, I ask them to empirically compare the performance of alternative designs (that instantiate alternative AI methods) along these dimensions. Then, I ask them to use this evaluation as a basis to reflect on the design process, and to critique the AI concepts and methods they used in their designs.

In general, students work on these projects in rotating teams of two or three. This enables the students to appreciate different perspectives on a problem, to articulate their own views, and to learn from one another. In the classroom, I use design in general as the context for motivating, introducing, illustrating and explaining basic AI concepts and methods. Outside the class, the Teaching Assistants and I work with the students in articulating design choices, observing the output behaviors of their systems, and analyzing the relationship between the design choices and the output behaviors.

Outcomes of the Design Stance

My analysis of the outcomes of these teaching goals and strategies are based on four main sources of information: the performance of students on the examinations and the design projects, the quality of class and individual discussions, student evaluation of their learning and my teaching, and, in one particular case, interviews with the students conducted by Georgia Tech's Center for Enhancement of Teaching and Learning. The main results are that students were better motivated, they better understood the significance of many AI concepts and methods and the connection between them, they enjoyed the design projects a lot, and they developed better design skills. However, the above teaching strategies seemed to have little or no impact on their learning of Lisp, and their experiments with the design projects did not enable them to critique AI concepts and methods probably because of insufficient scaffolding.

Tools for Assessment of Teaching

Clearly, the assessment of teaching and learning are critical issues in improving the quality of teaching and learning. Assessment of a student's learning enables the teacher to provide feedback which can help reformulate the student's learning goals and strategies. Similarly, the assessment of teaching can provide feedback to the teacher which may lead to a reformulation of the teaching goals and strategies.

The above analysis of the outcomes of my teaching strategy and methods is based on the standard methods and tools for assessing teaching and learning such as the performance of students on examinations and projects, and student evaluation of teaching, But these assessment tools are informal and imprecise at best. Thus, my analysis of the outcomes too is informal and imprecise.

Given that we want to enhance the quality of AI teaching, we need to develop methods and tools for assessing the teaching so that we can determine what worked, what did not and why, how can we improve the teaching, and for sharing this knowledge with other AI educators. The design stance towards teaching AI does not seem provide any help in this regard.

Recently, I have begun to use 'course portfolios' patterned after [Cerbin 1993] for self-assessment of teaching. A course portfolio apparently provides three main benefits. First, it forces the teacher to explicitly and precisely spell out the teaching goals and objectives, the teaching strategies and methods, and how the strategies are expected to accomplish the goals. While all teachers start with some goals and strategies, often they are implicit and tacit, and the relation between them is imprecise and ambiguous. Second, the course portfolio ties the assessment of teaching with the assessment of learning. If the teaching goals and strategies are clear and the relation between them is precise, and if the students' learning can be assessed to some degree, then the assessment of the students' learning also provides a partial assessment of the teaching. Third, the course portfolio is a written document, constantly evolving, and sharable with other educators.

Since I started developing and using course portfolios only recently, it is still too early to make any conclusions about the impact of a teaching strategy based on the portfolios. Nevertheless, my preliminary impression from the course portfolio I have developed for one introductory AI class [Goel 1994] appears to confirm the advantages of the design stance towards teaching AI.

References

Bill Cerbin. The Course Portfolio as a Tool for Continuous Improvement of Teaching and Learning. Paper presented at the *American Association for Higher Education Double Feature Conference: The 8th Annual Assessment Conference and The 1st Continuous Quality Improvement Conference*, Chicago, June 1993.

Ashok Goel and Todd Callantine. An Experience-Based Approach to Navigational Path Planning. In *Proc. IEEE/RSJ International Conference on Robotics and Systems*, Rayleigh, North Carolina, July 1992, IEEE Press, Volume II, pp. 705-710.

Ashok Goel, Michael Donnellan, Nancy Vasquez, and Todd Callantine. An Integrated Experience-Based Approach to Navigational Path Planning for Automonous Mobile Robotics. In *Proc. IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, May 1993, pp. 818-825

Ashok Goel. Lilly Project Report: Teaching Design Skills. Internal Memo, AI Group, College of Computing, Georgia Institute of Technology, October 1993.

Ashok Goel. A Course Portfolio for CS 6361, Graduate Introduction to Artificial Intelligence. Internal Memo, AI Group, College of Computing, Georgia Institute of Technology, February 1994.

Herbert Simon. *The Sciences of the Artificial.* Second Edition. Cambridge, MA: MIT Press, 1981.

Working Notes of the Cognitive Science Workshop on Teaching Design Skills, Georgia Institute of Technology, Atlanta, March 1992.