A Consideration of Some Approaches to Course Organization

S. Rebecca Thomas
Department of Computer Science
University of Northern Iowa
Cedar Falls, IA 50614-0507
thomas@cs.uni.edu

Abstract

An introductory artificial intelligence course may be taught with emphasis on theory and formalism, or with emphasis on engineering artifacts that display some aspect of intelligence. It makes pedagogical sense to organize the course around a central problem or approach, but the course may instead be taught as a smorgasbord of approaches and problems. This paper briefly considers the four possible combinations of approaches.

When designing introductory artificial intelligence courses, instructors must consider what material is appropriate for such a course and which pedagogical approaches are most likely to be successful. The choices to be made include at least

- Focusing on formal, theoretical approaches to problems vs. focusing on engineering approaches, and
- Organizing the course around central problems and approaches vs. considering a sampler of problems and approaches from different areas.

The first choice (formal vs. engineering approaches) is clearly a continuum; the proportion of out-of-class assignments that involves programming, as opposed to pencil-and-paper work such as proofs or analysis of problems, gives a good indication of where on this axis a particular course lies. Instructors must consider the mathematical sophistication and programming abilities of their students in deciding where on this continuum their course should fall; probably neither extreme is optimal. The second choice also lies along a continuum, of course, but here I think the extremes are more attractive than the middle. Many of the faculty who attended the 1994 NSF-sponsored workshop on Providing and Integrating Educational Resources for Faculty Teaching Artificial Intelligence agreed that material is generally easier for students to absorb and retain when it is clearly organized and related to some central theme. However, the workshop attendees also agreed that it is difficult to see what theme should be used for introductory artificial intelligence courses, perhaps due to the field's youth. Therefore, perhaps it is preferable to give up the idea of a central organizing theme, and to present instead the wide variety of approaches that are currently being explored in artificial intelligence. Consideration of these two ways of splitting the space of possible approaches to organizing courses gives us four "quadrants;" I will briefly consider each.

The formal-central approach, in which the course is organized around a central approach and in which a significant amount of emphasis is placed on formalism, with correspondingly less emphasis on implementation and experimentation (since time is limited), is one this author is familiar with through experience as a teaching assistant and as a student. Search seems a promising central theme for the course, since many problem-solving methods can be described as search through some solution space. Students would study and probably implement different search algorithms, and analyze their properties theoretically and perhaps experimentally. They would then learn about such areas as planning, concept formation, and theorem proving in terms of search through various kinds of spaces, and much attention will be paid to formalizing these problems. One problem with this approach is that it works best with so-called "core AI" material, and other areas (vision, for example) may get short shrift in such a course. This approach is probably best-suited to students with some degree of mathematical sophistication; it will not work as well with students who are uncomfortable with abstraction. (Although it may give them valuable *practice* at dealing with abstraction, this is likely to interfere with learning the course material.)

In the formal-sampler approach, on the other hand, emphasis would again be on theory and formalism rather than implementation, but different formalisms would be introduced for different areas of AI research. Students might study search, logics (perhaps including modal logics), optics, mechanics, decision theory, and so on. This approach seems likely to overwhelm all but the mathematically sophisticated, and seems likely to focus too much student energy on understanding formalisms rather than on understanding concepts. A course designed in this way might be very useful for advanced students who expect to become AI researchers,

but it seems inappropriate for an introductory course.

An engineering-central approach might be implemented in several different ways; I'll discuss two. One might, as in the formal-central approach suggested above, make search the theme of the course. Students would implement search procedures early in the semester, analyze them formally and test them experimentally, and then implement reasoning in different domains using search. An advantage of using this approach is that it would be natural to have students experiment with different knowledge representation strategies, and to evaluate them from the viewpoint of software engineering. In addition, this course is more accessible to mathematically unsophisticated students than a more formal approach. This approach suffers, however, from the problem mentioned above of focusing on "core AI" to the detriment of other areas.

Still under the engineering-central approach, one might focus on a central *problem* rather than a central *solution strategy*, exploring different approaches to (parts of) this central problem. The choice of the central problem is crucial. It must be difficult enough to keep students interested for the length of a term, and it should be amenable to several different approaches. This is the organizing principle I am currently using, with the example of a "gopher robot" along the lines of a twenty-first century Shakey. At the first class meeting we discussed the many problems that must be solved in order to build such a robot (navigation, natural language understanding and generation, probably vision, etc.) and throughout the semester I motivate each topic by relating it back to the problem of building this robot. Lab exercises involve vastly simplified versions of some of the problems faced by such a robot. It is premature to evaluate the success of this approach.

The engineering-sampler approach would introduce students to a number of small working systems that demonstrate successful approaches to a number of different kinds of problems. Students might build or explore a small expert system, a planner, neural nets, genetic algorithms, and so on. This approach seems most promising for fun laboratory assignments (it has a high "wow" potential), which should make students enthusiastic about pursuing further studies in artificial intelligence. Furthermore, it is easily adaptable to students' capabilities; advanced students can program from scratch, while less capable students experiment with and modify software provided. For those whose classes involve students with little programming experience, laboratory assignments are constrained to use canned software anyway. A danger of this approach is that very well-done software may unduly impress naive students, and instructors would have to be careful not to oversell the current state of the art. A difficulty is that the instructor has to spend time gatherizing and becoming familiar with many different software packages. Fortunately, resources like the Carnegie Mellon University AI Repository can help[1].

In choosing which of the approaches above makes sense in a particular situation, instructors must consider their own interests and preferences, the mathematical and computing sophistication of their students, the goals of the course, and the resources available for student and instructor use. Naturally there are many other ways to cut the space of approaches to teaching introductory AI, but the theory-engineering and central-sampler axes cut the space in illuminating ways and separate very different approaches to instruction.

Acknowledgments

My thanks to my colleagues at Northern Iowa for long discussions concerning pedagogy, and to the organizers of and participants in the NSF Workshop on Providing and Integrating Educational Resources for Faculty Teaching Artificial Intelligence. All of these people have contributed to my thinking on how to teach introductory AI, and how to teach in general. Thanks also to Mark Kantrowitz and Carnegie Mellon University for creating and maintaining the AI Repository.

[1] Via WWW, the AI Repository is found at

http://www.cs.cmu.edu:8001/Web/Groups/AI/html/repository.html.

Alternatively, files can be fetched via anonymous ftp; connect to ftp.cs.cmu.edu, do "cd /user/ai," and fetch readme.txt.