Augmenting AI Coursework Through Undergraduate Research

Antonio M. Lopez, Jr.

Xavier University of Louisiana
Computer Sciences and Computer Engineering Department
One Drexel Drive, Campus Box 50, New Orleans, LA 70125
tlopez@xula.edu

Abstract

All courses in Artificial Intelligence are not equal. The topics covered by a course entitled Artificial Intelligence vary widely. The *Computing Curricula 2001: Computer Science* offers a good deal of flexibility for degree programs to meet the prescribed standard of knowledge units for the field of Intelligent Systems. Most, but not all, Historically Black Colleges and Universities can achieve more than the minimum recommended core hours through a one-semester, intermediate-level course in the Junior/Senior year. A few have the ability to offer at least one advanced course as a Senior Elective. At our institution, we found undergraduate research projects to be an excellent means of preparing students for the one-semester, intermediate-level Artificial Intelligence course, or extending what can be covered in that course. This paper presents a review of the suggested coursework for a one-semester, intermediate-level course in Artificial Intelligence and what is possible at a non-Research Type I institution. It then gives two areas where undergraduate research projects have been used to create interest in or expand knowledge of Artificial Intelligence topics, thus covering more than what is possible in one course.

Introduction

In a 1998-99 survey of accredited undergraduate computer science degree programs (McCauley and Manaris 2000) with 82 out of 151 departments responding, only 6% required an Artificial Intelligence (AI) course for graduation. For 89% of the responding institutions, an AI course was a computer science elective. At the undergraduate level, some argue that a properly designed AI course would be accessible to students with good analytical skills though they might lack programming skills (Pfeifer 2000; Wyatt 2000). The *Computing Curricula 2001: Computer Science* (ACM 2001) has now recommended that there be a minimum of ten core hours devoted to knowledge units in the field of Intelligent Systems. Some undergraduate computer science departments will choose to embed these core hours into existing computer science courses. Others will embed them in interdisciplinary courses supported by other departments such as philosophy, and yet others will now require a one-semester, intermediate-level AI course for graduation. These determinations will more than likely be dependent upon the number of faculty willing to teach an AI course, the demands on faculty in other areas of the curriculum, or constraints such as time and equipment.

Historically Black Colleges and Universities (HBCUs) have traditionally struggled with curriculum issues due to a limited number of faculty and other constraints. Nonetheless, undergraduate students who engage in scientific investigation side-by-side with faculty at HBCUs have gained valuable experience enabling many of them to continue with graduate education at major research institutions (Benowitz 1997). Undergraduate research has been a driving force at our institution. The university has a Center for Undergraduate Research that focuses all the efforts being made by faculty and students in the Sciences, Technology, Engineering and Mathematics (STEM). Each year the Center sponsors the University's Showcase of Scholars where undergraduate researchers present poster sessions of their work, and the best projects receive awards. We have found this type of environment very nurturing for what we are trying to achieve in AI education.

AI Coursework

The *Computing Curricula 2001* stipulates that in the body of knowledge there will be a minimum of ten core hours (See Table 1) devoted to the field of Intelligent Systems (IS). These knowledge units can be embedded in courses other than those having AI as their primary focus. For example, the IS1 knowledge unit can be covered in an introductory-level programming course that uses a breadth-first approach, while IS2 might be covered in a data structures course and IS3 in a software engineering course. Or the entire core IS knowledge units can be together in an intermediate-level course as suggested in *Computing Curricula 2001*: CS260 Artificial Intelligence (System-based or Traditional approach), CS261 AI and Information (Web-based approach), or CS262 Information and Knowledge Management (Compressed approach). Each of the three courses takes a minimum of 40 classroom hours to present. The CS262 course has the three IS core knowledge units along with some core knowledge units from Information Management, Human-Computer Interaction, and Social and Professional Issues. It also has non-core knowledge units from Algorithms and Complexity as well as Net-Centric Computing. The CS261 course has the three IS core knowledge units and an additional two non-core IS knowledge units. It also has both core and non-core knowledge units in Information Management and one core hour in Social and Professional Issues, but it offers seven hours of time for elective topics. The CS260 course contains the entire core and non-core IS knowledge units and gives six hours for elective topics.

IS1: Fundamental intelligent systems issues (1 hr)	History of AI; philosophical questions; fundamental definitions; modeling the world; the role of heuristics
IS2: Search and constraint satisfaction (5 hr)	Problem spaces; brute-force search; best-first search; two-player games; constraint satisfaction
IS3: Knowledge representation and reasoning (4 hr)	Review of propositional and predicate logic; resolution and theorem proving; non-monotonic inference; probabilistic reasoning; Bayes theorem

Table 1. Core IS knowledge units.

The CS260-like intermediate courses in AI that are offered in colleges and universities throughout the United States vary in the topics covered and the depth to which those topics are covered (Kumar 1999) even if the same textbook is used. One of the authors of a respected AI textbook (Russell and Norvig 1995) has a Web Page entitled *Introductory AI Courses* (Russell 2003) that lists some of the institutions of higher education that are using the text as well as the chapter sequences that they employ. These variations are due in part to the underpinning of the course (i.e., general introduction, natural language, robotic, etc.) and the amount of time available (i.e., quarter or semester). It is interesting to note that the institutions posted on the site are, for the most part, Type I Research Institutions and few would be classified as liberal arts institutions. Furthermore, none of the institutions on the list are HBCUs.

Most HBCUs are Type II A or B Educational Institutions with a majority having their roots in the liberal arts. Nonetheless, many offer strong programs in computer science and computer information systems, and some have computer engineering programs. However, in the world of HBCUs it is not assured that students are afforded the opportunity of taking an intermediate course in AI let alone an advanced course in the subject. There are approximately 90 HBCUs. A recent random sampling of these institutions revealed that out of 18 institutions 12 had a one-semester, intermediate-level AI courses and 2 of the 12 also had at least one advanced AI course. Most of the intermediate AI courses were CS260-like in their description. Six HBCUs Computer Science programs did not have any AI courses listed in their course offerings. Of two HBCUs having computer engineering programs, the AI course was found in the computer science program. It is not clear, based upon the available catalog descriptions of the intermediate AI courses, how far beyond IS1, IS2, and IS3 the courses actually go.

Our HBCU was not among those in the quick survey that we conducted. Our computer science program takes a more traditional approach, and we have both an intermediate AI course and an advanced course in Neural Networks. We are fortunate to have four out of ten faculty members interested or actually doing research in AI. A faculty member who works in the AI field also teaches the Sophomore-level Software Engineering course. The text (Hamlet and Maybee 2001) for this course has a section in formal methods that uses the PROLOG programming language. Knowledge representation, searching, and, of course, understanding predicate logic, resolution and theorem proving are thus introduced in the Software Engineering course. Those students taking this course are exposed to these AI concepts well before they take our intermediate AI course. Because of high personal interest after this course, some of the students have gone directly into an undergraduate AI research project. Faculty members are willing to work with such students not only involving them in an AI research project but also preparing them for the AI course. We note that faculty at other institutions have seen the value of integrating AI and Software Engineering projects (August 2003; Pedrycz 2002).

With respect to an intermediate AI course, our students are not like those at Rice University or Yale University that can endure an instructor covering 21 chapters out of *Artificial Intelligence: A Modern Approach* (Russell and Norvig 1995) by working on course material many extra hours outside the classroom during the semester. Many of our students have "part-time" jobs to help pay for their college education and this limits the amount of time that they can spend studying for one course in one semester. So the coverage for us is more like 10 to 12 chapters. This means that we are barely able to get into AI planning. Our coverage in any depth of machine learning, perception, and building agents is out of the question in this one course, although some of the topics such as genetic algorithms or simulated annealing (from IS4) can be covered in our advanced course, Neural Networks. However, what we are able to cover in our one-semester, intermediate-level AI course is enough to spark an interest that can lead to undergraduate research in AI. Undergraduate research can be done over a period of time that is longer than a semester. In fact, our students often get summer employment, grants, or stipends to work on their undergraduate research projects. In sum, our better students get the full coverage of the IS body of knowledge, just not in one semester. We should mention too that undergraduate research projects are not done for credit hours; they are done because our students want to do them.

Ontology Development

Ontology development and use is an important area of research in AI (Swartout and Tate 1999), and we have found it to be a fertile area for undergraduate research. Ontology development is something that our undergraduates can do prior to taking our AI course. Ontologies provide the vocabulary that is common to knowledge workers in a particular problem domain. An ontology explicitly describes the different concepts and the relationships that exist between concepts thus giving structure for the knowledge. The graphical techniques employed in ontology development (See Figure 1) are well within the comprehension and reasoning abilities of an undergraduate who has not had an AI course. The nodes are concepts and the lines connecting the nodes are relationships.

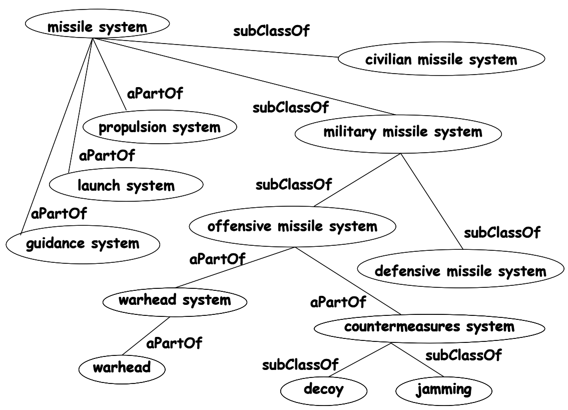

Figure 1. An ontological fragment about missile system.

A formal definition of an ontology is a logical theory, which gives an explicit, partial account of a conceptualization; it is an intentional semantic structure that encodes the implicit rules constraining the structure of a piece of reality (Guarino and Giaretta 1995). Every knowledge model has an ontological commitment; this is to say, that every knowledge model has a partial account of the intended conceptualization of a logical theory (Noy and Hafner 1997). Ontologies are essential for developing and using knowledge-based systems.

There are several types of ontologies (Maedche 2002). Top-level ontologies describe very general concepts, which are independent of a particular problem or domain (e.g., space, time). Domain ontologies specialize the concepts introduced in a top-level ontology, describing the vocabulary related to a generic domain. Task ontologies specialize a top-level ontology, focusing on a generic task or activity. Application ontologies provide concepts corresponding to roles played by domain entities while performing a particular activity.

Because of the software engineering influence and a desire not to complicate the undergraduate research more than necessary, our students have used the Skeletal Methodology (Uschold and King 1995) to develop their application and task ontologies. The Skeletal Methodology provides the following guidelines for the development of an ontology: (1) Identify the goal and the intended use of the ontology. (2) Identify the concepts and relationships between concepts that will comprise the ontology. Along with this activity it is useful to develop a glossary of terms, which in a machine-human partnership will allow the human to ask, "What is a countermeasures system?" and the machine to respond with the explanation, "Measures such as decoys, jamming, and chaff taken by an attacker to deceive a missile defense system." (3) Represent the acquired information in a formal language. Our undergraduate researchers are using PROLOG. (4) Import existing ontologies that are useful. (5) Evaluate the ontologies using competency questions -- i.e., queries to the ontology for which the response is known and can be compared to the machine's response. (6) Document the scope and depth of the ontology. With the exception of step 4, our undergraduate researchers have been able to use this methodology in developing their application ontologies. One of our undergraduate researchers is currently working on a task ontology that will import the ontological work done by another undergraduate researcher. Trying to understand the problems surrounding the importation of an ontology is a courageous step for an undergraduate to take. Although our undergraduate researchers have been focused on ontology development, they have done so with an eye towards building intelligent agents that can make use of the ontology.

Ontology development has either introduced students to or reinforced their understanding of topics in the core IS knowledge units, specifically from IS1 the modeling of the world and fundamental understanding of optimal vs. human-like reasoning, from IS 2 the topics of problem spaces and brute force searching as well as constraint satisfaction, and from IS3 propositional and predicate logic, resolution, and theorem proving. The work also introduced topics from the non-core IS knowledge units, from IS5 structured representations such as frames and description logics as well as the topic of inheritance systems, and from IS9 the defining of planning systems, case-based reasoning, and static world planning.

Finding a domain in which to have the student work has not been a problem. For example, the various violent acts being perpetrated globally in the name of a religion motivated one computer science undergraduate researcher to begin developing an ontology on world religions before she had taken our AI course. The student has a deep interest in her own religion and wanted to understand how other religions that have common origins, could be in such conflict with one another. Religions form a psychological context in which some people function violently, for example, the Kamikaze functioned violently in the context of Bushido, and the Hamas suicide bombers function in the context of Islam. But Bushido and Islam are not as well related as say Islam and Judaism, both of which trace historical paths from Abraham. Religions also form a social structure within society; there are religious leaders (rabbi, imam, bishop, etc.), buildings (synagogue, mosque, church, etc.), and other trappings. After a year and a half of work, the competency queries given to the PROLOG implementation of her ontology have demonstrated that the ontology needs more work before an agent can use it successfully. Nonetheless, the student presented a poster session at the University's Showcase of Scholars and won second prize in a student paper competition at a regional computer science conference (Gilds 2002). Furthermore, she was recognized for her work as the runner-up in the competition for the Greater New Orleans Sigma Xi Gold Medal Award, given for outstanding, original undergraduate research in a STEM discipline.

A computer engineering undergraduate researcher was interested in various kinds of sensors and the knowledge required to make intelligent use of them. In particular, she was interested in active radar sensors. She approached a faculty member working on a Domain ontology for missile defense. She proposed to build a sensor ontology that might be incorporated into the faculty member's ongoing research. The faculty member was able to encourage the work and hire the student for the summer. This student had not taken either the Software Engineering course or our AI course. While collecting information on the concepts and relationships that would be useful, she learned PROLOG. Her ontology grew quickly and had to be reorganized several times as new concepts and relationships were discovered. She made excellent use of competency questions to improve her ontology. In Spring 2003, she presented her research at the University's Showcase of Scholars. She was awarded first prize. This encouraged her so much that she "volunteered" to place her poster presentation in competition with graduate students at a national conference. Although she did not win an award in this competition, she learned a great deal about graduate students. She has inspired other students at our university to pursue undergraduate research. Presently, she and the student who is working on the task ontology are preparing a joint paper for submission to a regional conference.

Semantic Web Agents

After students have taken our AI course, we invite them to either continue their undergraduate research projects or to get involved in an existing AI undergraduate research project. However, we now shift the focus from ontology development to building intelligent agents that will use ontologies. We have found that students are inclined to accept this invitation if we associate the work with ongoing Semantic Web research. Ontologies are essential for enabling the Semantic Web (Davis, Fensel, and van Harmelen 2003), and this provides our other venue for undergraduate research.

The Semantic Web is the envisioned end-state for the migration of the World Wide Web from words, images, and audio understood only by humans to those same things "cloaked" in ontological structures understood by both humans and intelligent agents (Berners-Lee, Hendler, and Lassila 2001). The World Wide Web Consortium (W3C) has been moving the current Web slowly toward the Semantic Web end-state. Two important steps were the extensible markup language (XML) and the resource description framework (RDF). On the Semantic Web, many people and organizations around the world will be developing and using ontologies to convey the meaning of their Web pages. The real power of the Semantic Web will be realized as intelligent agents are constructed to use these ontologies to collect Web content from diverse sources, process the information, and exchange the results with other intelligent agents and humans.

An intelligent agent is a knowledge-based software system that is capable of (1) perceiving its environment, (2) determining reasoned courses of action by interpreting perceptions, drawing inferences and solving problems in that environment, and (3) acting upon the environment to realize a set of goals and tasks for which it was designed (Russell and Norvig 1995). The definition of agents, their architecture and theory as well as their implementation are topics in the non-core IS6 Agents knowledge unit. The agents that we want our undergraduate researchers to develop are information-gathering agents, also an IS6 topic.

We make the transition from the ontology work and the building of agents by connecting the ontology to the markup languages that the agent must use. Presently, XML allows creators of Web pages to produce and use their own markup tags. If other users of the Web pages know the meaning of the tags, then they write scripts that make use of the tags. Unfortunately, XML does not provide standard data structures and terminologies to describe processes and the exchange of information (Davis, Fensel, and van Harmelen 2003). So as far as an intelligent agent is concerned, understanding has not been sufficiently advanced with XML; thus the need for RDF. The RDF data model, which is equivalent to the graphic formalism illustrated in Figure 1, consists of resources, properties, and statements written using XML tags. Universal Resource Identifiers (URIs) can be used in conjunction with Universal Resource Locators (URLs) to identify resources, properties, and statements. In addition, the RDF Schema (RDFS) provides a means of defining relationships between resources and properties. Hence, the RDFS provides the basics for defining knowledge models that are similar to frame-based systems. Figure 2 illustrates these connections.

The Defense Advanced Research Projects Agency (DARPA) has also supported the development of the Semantic Web. It has funded research in languages, tools, infrastructure and applications. The DARPA Markup Language (DAML) was developed and coupled with the Ontology Inference Layer (OIL) to produce DAML + OIL, a proposed starting point for W3C's Web Ontology Language (OWL). Web integration, frame-based systems, and description logics inspired the specification of OWL (McGuinness et al. 2002).

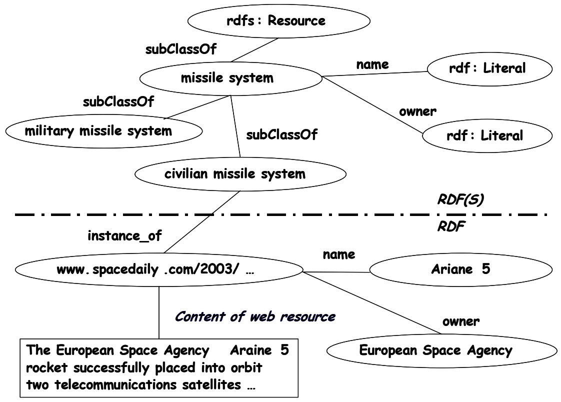

Figure 2. Connecting ontology to Semantic Web markup.

The encoding of a Web page using XML, RDF, RDFS, and DAML + OIL is straightforward, but still subject to change as W3C continues its research work on OWL. The underpinning is without doubt the ontological structures that support the Web page. Table 2 gives a glimpse of the markup language implementation of Figure 2. In Table 2 two ideas that are tacit in Figure 2 have been made explicit. The first idea is that missile systems are either military missile systems or civilian missile systems (disjointness -- `disjointUnionof`). The second idea is that the concept of a missile system is the same as the concept of a rocket (sameness -- `sameClassAs`).

The stage has been set for building intelligent agents that will gather and use information from Semantic Web pages. Building intelligent agents is not an easy matter for undergraduates. However, our students are willing to work toward this goal due in part to what they have already accomplished and in part to what they are reading in the current periodical literature (Tsai et al. 2003; Shanks, Tansley, and Weber 2003). They see their research as something real and useful; something that might help them get into graduate school or find employment after college in a career field that truly interests them. They also understand that even though they had our AI course there is still a great deal more to learn; we are not able to cover in any great detail the architecture and theory supporting agent technology. This is accomplished primarily as undergraduate research. In the end, we are guiding our undergraduate researchers toward being life-long learners. What more could an educator want?

Conclusion

Computing Curricula 2001 has provided a great deal of flexibility in teaching AI. HBCUs need this kind of flexibility to insure that their students get the core IS knowledge units. Although we are fortunate enough to be able to support a one-semester, intermediate-level course in AI and one advanced AI course, we want our students to learn more. We have found undergraduate research projects to be excellent way of accomplishing our goal. This paper presented two areas that are appropriate for such work – ontology development before students have taken our AI course and development of Semantic Web agents after the students have taken the AI course.

```
<?xml version="1.0" ?>

<rdf:RDF

xmlns:rdf=
"http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs=
"http://www.w3.org/2000/01/rdf-schema#"
xmlns:daml=
"http://www.daml.org/2001/03/daml+oil#">
…
<daml:Class rdf:ID="missile_system">
<rdfs:label>missile_system
</rdfs:label>
</daml:Class>

<daml:Class rdf:ID=
  "military_missile_system">
<rdfs:subClassOf rdf:resource=
  "#missile_system"/>
</daml:Class>
…
<daml:Class rdf:ID=
  "civilian_missile_system">
<rdfs:subClassOf rdf:resource=
  "#missile_system"/>
</daml:Class>

<daml:Class rdf:about="#missile_system">
<daml:disjointUnionOf rdf:parseType=
  "daml:collection">
<daml:Class rdf:about=
  "#military_missile_system"/>
<daml:Class rdf:about=
  "#civilian_missile_system"/>
</daml:disjointUnionOf>
</daml:Class>

<daml:Class rdf:ID="rocket">
<daml:sameClassAs rdf:resource=
  "#missile_system"/>
</daml:Class>
…
</rdf>
```

Table 2. Ontology markup.

References

ACM 2001. *Computing Curricula 2001: Computer Science*. See http://www.computer.org/education/cc2001/final/

August, S. 2003. Integrating Artificial Intelligence and Software Engineering: An effective interactive AI resource does more than teach AI. In Proceedings of the 2003 Information Resources Management Association

International Conference, 17-19, Philadelphia, PA: Information Resources Management Association.

Benowitz, S. 1997. Historically Black Colleges Combine Research, Education. *The Scientist* 11(4):14-17.

Berners-Lee, T.; Hendler, J.; and Lassila, O. 2001. The Semantic Web. *The Scientific American* 284(5):34-43.

Davis, J.; Fensel, D.; and van Harmelen, F. eds. 2003. *Toward the Semantic Web: Ontology-driven Knowledge Management*. West Sussex, England: John Wiley and Sons Ltd.

Gilds, B. 2002. Knowledge Engineering: Ontology of World Religions. *The Journal of Computing Sciences in Colleges* 18(1):304-308.

Guarino, N.; and Giaretta, P. 1995. Ontologies and Knowledge Bases: Toward a Terminological Glarification. In *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, ed. N. Mars, 25-32, Amsterdam, The Netherlands: ISO Press.

Hamlet, D.; and Maybee, J. 2001. *The Engineering of Software: Technical Foundations for the Individual*. Boston, MA: Addison Wesley Longman, Inc.

Kumar, D. 1999. Curriculum Descant: Beyond Introductory AI. *Intelligence* 10(3):13.

Maedche, A. 2002. *Ontology Learning for the Semantic Web*. Norwell, MA: Kluwer Academic Publishers.

McCauley, R.; and Manaris, B. 2000. An Information Resource for Computer Science Educators. *ACM SIGCSE Bulletin: inroads* 32(2):25-29.

McGuinness, D.; Fikes, J.; Hendler, J. and Stein, J. 2002. DAML + OIL: An Ontology Language for the Semantic Web. *IEEE Intelligent Systems* 17(5):72-80.

Noy, N.; and Hafner, C. 1997. The State of the Art in Ontology Design: A Survey and Comparative Review. *AI Magazine* 18(3):53-74.

Pedrycz, W. 2002. Computational Intelligence as an Emerging Paradigm of Software Engineering. *Proceedings of the Fourth International Conference on Software Engineering and Knowledge Engineering*, 7-14. New York, NY: ACM Press.

Pfeifer, R. 2000. Teaching "New AI". *Intelligence* 11(2):17-19.

Russell, S. 2003. *Introductory AI Courses*. See http://www.cs.berkleley.edu/~russell/courses.html.

Russell, S.; and Norvig, P. 1995. *Artificial Intelligence: A Modern Approach.* Upper Saddle River, NJ: Prentice Hall.

Shanks, G.; Tansley, E. and Weber, R. 2003. Using Ontology to Validate Conceptual Models. *Communications of the ACM* 46(10):85-89.

Swartout, W.; and Tate, A. 1999. Ontologies. *IEEE Intelligent Systems* 14(1):18-19.

Tsai, T.; Yu, H.; Shih, H.; Liao, P; Yang, R. and Chou, S. 2003. Ontology-mediated Integration of Intranet Web Services. *Computer* 36(10):63-71.

Uschold, M., and King, M. 1995. Towards a Methodology for Building Ontologies. Paper presented at the Workshop on Basic Ontological Issues in Knowledge Sharing at the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95), 20-25 August, Montreal, Canada.

Wyatt, R. 2000. Interdisciplinary Artificial Intelligence. *Intelligence* 11(1):11-12.