Design of an Online Course on Knowledge-Based AI

Ashok K. Goel and David A. Joyner
Design & Intelligence Laboratory, School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA 30338, USA
ashok.goel@cc.gatech.edu; david.joyner@cc.gatech.edu

Abstract

In Fall 2014 we offered an online course on Knowledge-Based Artificial Intelligence (KBAI) to about 200 students as part of the Georgia Tech Online MS in CS program. By now we have offered the course to more than 1000 students. We describe the design, development and delivery of the online KBAI class in Fall 2014.

Georgia Tech OMSCS Program

In January 2014, Georgia Tech launched its Online MS in CS (OMSCS) program (www.omscs.gatech.edu/). The video lessons for the OMSCS courses are delivered by Udacity (www.udacity.com/georgia-tech). The goal is to offer the same educational programs and courses online that we offer to residential MS students, and with the same depth, substance and rigor. The OMSCS program currently has ~3000 students, which is an order of magnitude more than the number of students in the residential program. However, while the residential degree costs several tens of thousands of dollars, the OMSCS program costs only several thousand dollars, an order of magnitude less costly than the residential program.

Georgia Tech KBAI Class

Goel has been teaching a residential semester-long course on Knowledge-Based AI at Georgia Tech each year for more than a decade. Joyner took the KBAI course in Fall 2010 and was a teaching assistant (TA) for the course in 2012. The KBAI class focuses on the cognitive systems school of AI that we characterize as *human-level*, *human-like* and *human-centered AI* (Goel & Davies 2011). The KBAI class adopts a *design stance* towards learning about AI (Goel 1994). Thus, the class work includes intensive design and programming projects that build on one another. In recent years, the class projects in the KBAI course have focused on visual analogy problems inspired by the Raven's Progressive Matrices (RPM) test of intelligence (Raven, Raven & Court 1998). The RPM test has attracted much interest in cognitive systems research (Bringsjord & Schimanski 2003; Carpenter, Just & Shell 1990; Lovett et al. 2009) including in our research laboratory (Kunda, McGreggor & Goel 2013; McGreggor, Kunda & Goel 2014). In the KBAI class, students design, program, and test AI agents on visual analogy problems inspired by the Raven's test. We found that the class projects stimulated student engagement while providing an authentic opportunity to explore cutting-edge research (Goel, Kunda, Joyner & Vattam 2013).

Design of an Experiment in Online Learning

We made several strategic decisions when we first conceived the online KBAI course. (1) We decided to view the design, development and delivery of the online KBAI course as an experiment in design-based research on online learning. (2) We decided not to directly transfer the course materials from the legacy residential KBAI class to the new online KBAI course. Instead, we viewed the task of designing the online course as an opportunity to reflect on the learning goals, strategies, outcomes, and assessments of the course. (3) We decided not to follow the most common method for making online courses: replay of videotapes of residential classes. We thought that this method is both limited by the constraints of the old classroom medium and takes minimal advantage of the affordances of the new online medium. (4) We decided to incorporate as many lessons from the learning sciences into the design of the online KBAI course as possible. Thus, we adopted learning strategies such as learning by example, learning by doing, project-based learning, collaborative learning, and more. (5) We decided to use as much interactive educational technology in the online course as possible. Thus, we developed and embedded ~150 interactive "microexercises" in the video lessons and ~100 AI agents as "nanotutors" into the exercises.

Learning Goals and Strategies

Our design for the online KBAI class follows a four-tiered learning hierarchy consisting learning goals, outcomes, assessments, and strategies.

Learning Goals

We have four main learning goals for the class. *G1 - Methods:* Students will learn the core methods of KBAI. *G2 - Tasks:* Students will learn the common tasks addressed by KBAI. *G3 - Systems:* Students will learn ways AI agents can use these methods to address these tasks. *G4 - Cognition:* Students will learn the relationship between KBAI and cognitive science, using theories of human cognition to inspire design of AI agents, and using the designs of the AI agents for insights into cognition.

Learning Outcomes

We expect three main learning outcomes based on the above learning goals: *O1 - Build Systems:* The primary learning outcome is that students should be able to design, implement, evaluate, and describe KBAI agents. *O2 - Address Complex Problems:* Students should also be able to use these strategies to address practical problems. *O3 - Reflect on Cognition:* In addition, students should be able to use the design of KBAI agents to reflect on human cognition (and vice versa).

Learning Assessments

We use main five types of assessments. *A1 - Projects:* Four design and programming projects. The four projects are related, with each project building on preceding projects. *A2 - Assignments:* Eight short assignments. In these written assignments, students will conceptually describe how a particular method might be used to complete the project. *A3 - Tests:* Two take-home tests, a midterm and a final, which examine students ability to reason through the application of the course's topics to a greater variety of problems than is covered in the project. *A4 - Exercises:* A number microexercises embedded in the video lessons. Although these are not incorporated into their grades, they provide us with a look at how students are interacting with and mastering the class material. *A5 - Interactions:* Students' interactions with one another, the TAs and the professor. Again participation is not graded explicitly, but it is explicitly set as an expectation at the beginning of the course.

Learning Strategies

We use ten broad pedagogical strategies: *S1 - Learning by Example:* Each of the ~25 lessons begins with an example of a real-world task for which we want to build an AI agent. This example is then used throughout the explanation of the method in that lesson to tie the method back to a particular practical problem. *S2 - Learning by Doing:* Each lesson includes several micro-exercises, one for each main concept in the class, for a total of ~150 microexercises over ~25 lessons. As students address each of the exercises, they are given targeted feedback directly to the nature of their answer. To give this feedback, we have constructed nanotutors for many exercises. *S3 - Authenticity of Learning:* Whenever possible, we take examples from the real world. Even when this is not possible, we relate the examples to the real world. *S4 - Project-Based Learning:* During the course, each student completes a semester-long project broken into four phases. The big project addresses a real, big and complex problem: taking an intelligence test. *S5 - Personalized Learning:* Personalization is incorporated throughout the course. First, on every exercise, students are given individualized, targeted feedback (S2). Similarly, on the projects, students are able to run their projects and receive feedback in real time on its success and can revise the project accordingly. *S6 - Collaborative Learning:* We form small "study groups" of the students in the course. While the tests, the projects and the assignments in the course require individual work, we encourage the study groups to work together on all aspects of the course (including discussions about the projects and the exercises). *S7 - Peer-to-Peer Learning:* After each test, project and assignment, we publicly post the best tests/projects/assignments along with our critiques. Students are requested to read through the exemplary work, and expected to raise their own work to the same level of excellence. *S8 - Learning by Teaching:* We empower the students and provide opportunities to students to act as teachers to one another. We ask the students to provide feedback to other students in their assignments. *S9 - Learning by Reflection:* At the conclusion of each lesson, we ask each student to reflect on what they learned in the class. Each design project requires the writing of a design report that explains and critiques, and reflects on the student's work on the project. *S10 - Community of Practice:* We use an online discussion forum dedicated to the class to help develop a community of practice. We encourage all students to introduce themselves on the forum, and support information sharing, question answering, as well as discussions. The teaching staff not only monitors the forum and

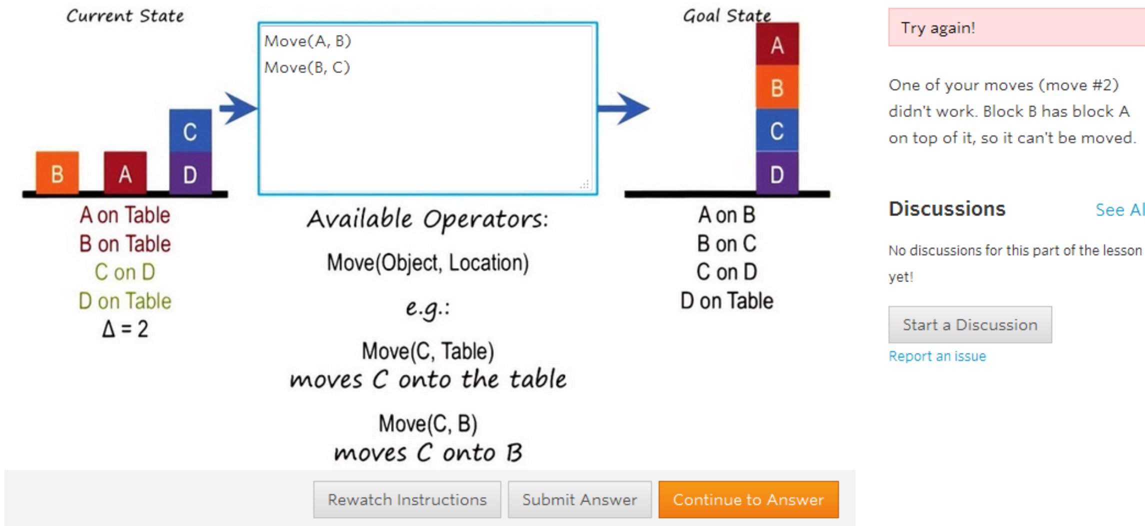

Figure 1: An example of a microexercise on the left and a nanotutor on the right.

publicly answers questions, but it also seeds discussions. We also hold regular office hours via Google Hangout.

Design of the Online Course

The online KBAI course comprises of 26 lessons on the following topics: (1) Introduction to the course, (2) Introduction to KBAI, (3) Semantic Networks, (4) Generate & Test, (5) Means-Ends Analysis and Problem Reduction, (6) Production Systems, (7) Frames, (8) Learning by Storing Cases, (9) Case-Based Reasoning, (10) Incremental Concept Learning, (11) Classification, (12) Logic, (13) Planning, (14) Understanding, (15) Commonsense Reasoning, (16) Scripts, (17) Explanation-Based Learning, (18) Analogical Reasoning, (19) Generalization and Version Spaces, (20) Constraint Propagation, (21) Configuration, (22) Diagnosis, (23) Learning by Correcting Mistakes, (24) Meta-Reasoning, (25) Advanced Topics, and (26) Course Wrap-Up. The lessons vary in length based on the topic (one of the advantages of preparing the class in this medium), but average to approximately one hour per lesson when including the time students spend completing the interactive microexercises in each lesson. The videos of all 26 lessons are now available publicly and freely through Udacity at https://www.udacity.com/course/knowledge-based-ai-cognitive-systems--ud409.

Exercises and Tutors

Embedded in the 26 video lessons are ~150 interactive microexercises, averaging to approximately six exercises per lesson. This leads to an interactive mircoexercise approximately every eight minutes in a lesson. Figure 1 illustrates one mircoexercise. The input for the exercise is comprised of free-response text that is instructed to follow a certain format. Other exercises in the course combine several multiple-choice questions, multiple free-response text boxes, and other more complicated structures.

The majority of the interactive exercises in the course are equipped with nanotutors that build on our work on intelligent tutoring systems (Joyner & Goel 2015). These nanotutors give students targeted, individualized, just-in-time feedback on students' responses to the current exercise. The tutors operate first by examining whether the input to the problem even makes sense. If not, the nanotutor supplies feedback on the type of input it will understand, guiding students along to the closed input set that it can process. Then, once it understands the input, it examines whether that input is valid; in the exercise above, it would check if all the moves are legal. If the input is valid according to the rules of the exercise, it moves on to checking correctness; in the exercise above, there exists valid feedback that does not answer the actual question of the exercise. Finally, for some microexercises, the nanotutor checks to see if the answer is the *best* answer. In the exercise above, the nanotutor may comment that while the goal was achieved, it could have been achieved in fewer moves.

Readings

The recommended readings came from several textbooks, including Winston (1993), Stefik (1995), Nilsson (1998), and Russell & Norvig (2009). In particular, the course covered about three fourths of the Winston book (most all, except for the chapters on search, genetic algorithms, neural networks, and vision), about a third of the Stefik book (especially the chapters on classification, configuration and diagnosis), and selected chapters from the Russell & Norvig book (such as planning). In addition, we included

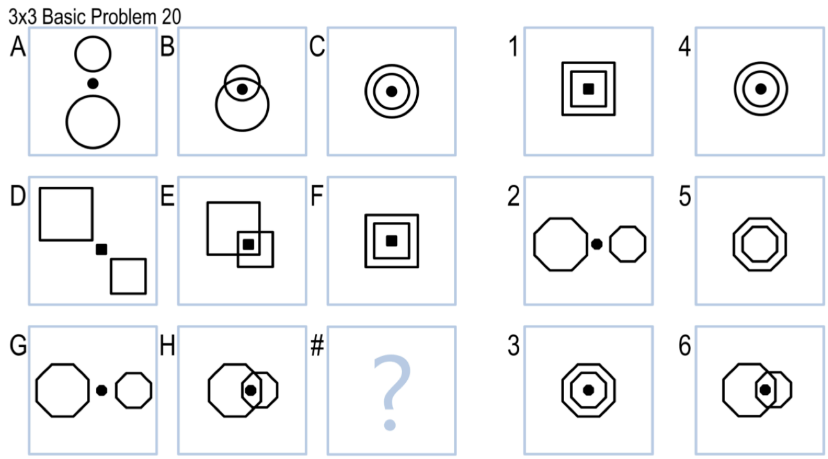

Figure 2. A 3x3 visual analogy problem inspired by the Raven's Progressive Matrices test. (The correct answer is 3.)

several optional readings on selected topics in cognitive systems such as Lehman, Laird & Rosenbloom (2006) on the SOAR cognitive architecture.

While the course does not teach AI programming, it provides access to AI programming resources such as the reimplementation of several classic AI systems in Python (Connelly & Goel 2013) described in (Norvig 1992).

Projects

The projects in the online KBAI class are built around the Raven's Progressive Matrices (RPM) test of intelligence. Due to copyright and other issues, we are unable to use the actual RPM as part of the class projects, but instead we have developed a set of RPM-inspired problems that leverage the same transformations and reasoning strategies seen on the actual RPM. Figure 2 illustrates a 3x3 problem from our problem set. On the left is a 3x3 matrix with one entry missing. On the right are six choices. The task again is to write an AI agent that can autonomously select one of the six choices on the right to insert into the missing entry on the left and thereby complete the pattern in the 3x3 matrix. For the Fall 2014 section of the class, we used 123 of such problems: 27 2x1, 48 2x2, and 48 3x3. Although 2x1 problems are not present on the actual RPM, we use them as a soft introduction to the type of reasoning that is needed on the test. In Fall 2014, students completed four projects. In projects 1, 2, and 3, students addressed the 2x1, 2x2, and 3x3 problems respectively; each project was also run against the problems from the previous project(s). In these three projects, students designed KBAI agents that operated on symbolic, verbal descriptions of the RPM problem. In Project 4, the input to the AI agents was the image files representing each frame of a problem. Goel & Joyner (2015) provide more information on these projects.

Assignments

In Fall 2014, students also completed eight written assignments. Each assignment had the same general prompt: choose any of the topics covered in the class and discuss how that topic might be used to address RPM problems. 24 total topics are covered in the class, meaning that each student would choose 8 of the 24 topics to use at some point during the semester. Early in the semester, these assignments served to help students brainstorm and gather feedback on their approaches to designing their agents; later in the semester, these assignments served to help students think about these techniques could be used to address bigger, broader problems than the handful of RPM provided during the projects.

Examinations

There were 2 examinations: 1 mid-term examination and 1 final examination. Both examinations were of the take home type. Thus, the students had access to all kinds of information resources at their disposal. All questions on the mid-term and final examinations were open-ended. For the mid-term examination, the questions were based on a science fiction story. The questions on the final examinations originated from research projects in our laboratory.

Development of the Online Course

'Development' here refers to the process by which the course materials were assembled, recorded, edited, and reviewed. The result of the development process is a kind of "video textbook" of sorts, a collection of high-quality video lessons together with the readings, projects, assignments, grading rubrics, and other reusable materials.

Development of the KBAI course began in February of 2014 with an intense two-day boot camp at Udacity. During this boot camp, we developed the learning goals, outcomes, and strategies for the entire course, as well as the entire structure of the course, identifying the 26 distinct lessons for production and recording.

After completing the boot camp, we began a two-month (March to May 2015) process of scripting the 26 lessons. Just as we had done for the course as a whole, for each lesson we articulated a set of learning goals, outcomes, and strategies, as well as a set of assessments and a lesson plan. Each lesson was constructed around a series of interactive microexercises.

At the conclusion of the scripting process, we spent two months (May to July 2015) recording the lessons and turning the scripts into polished, final videos. After recording all the filmed material for the class, we assembled the descriptions and rubrics for the course's four projects, eight assignments, and two exams (July to August 2015).

Delivery of the Online Course

'Delivery' here refers to the act of actually teaching the course in a particular semester, which involves several facets that cannot be reused from semester to semester, such as office hours, virtual interactions on the discussion forums, examinations, and the actual grading. The KBAI course launched in middle of August 2014 and ended in the middle of December 2014 for a total of 16 weeks of learning. We used a number of tools during the course: videos lessons were delivered via Udacity, assignments and announcements were given and received through Georgia Tech's Learning Management System T-Square; office hours were handled through Google Hangouts; and the discussion forum was hosted on Piazza (www.piazza.com). We also an interactive tool developed locally by our colleague Joe Gonzales, a graduate student at Georgia Tech, for managing peer-to-peer feedback on the assignments. In addition, we wrote scripts for running and grading the students' AI agents on the 123 problems in the four projects.

Evaluation of the Course

In evaluating the online course, we took two approaches. First, we looked at student responses to the several surveys offered during and at the end of the course. Overall, students were overwhelmingly positive about the course. One student commented, "Please have other OMSCS courses follow the teaching methodology used in this course." Another replied, "Overall, one of the best courses I have ever taken either in person or online." And, perhaps most significantly, a third wrote, "This course impressed on me so much that I have changed my specialization from Software and DB to Interactive Intelligence." Goel & Joyner (2015) provide details.

Table 1. Average grades on each assignment for the residential and online sections of CS7637 in Fall 2014.

Item	Max	OMSCS (Mean)	Residential (Mean)
Assignment 1	4	3.90	3.52
Assignment 2	4	3.94	3.70
Assignment 3	4	3.95	3.52
Assignment 4	4	3.92	3.83
Assignment 5	4	3.89	3.75
Assignment 6	4	3.86	3.62
Assignment 7	4	3.91	3.77
Assignment 8	4	3.97	3.90
Project 1	100	94.47	92.61
Project 2	100	92.74	89.64
Project 3	100	93.10	92.17
Project 4	100	92.0	88.5
Midterm	100	70.2	70.0
Final Exam	75	93.76	93.48
Final Grade	100	92.32	91.31

Second, we looked at student performance in the course, especially in comparison to the residential course. As Table 1 indicates, OMSCS students outperformed residential students on every assessment in the class and in the class as a whole. Students in the Fall 2014 offering of the KBAI course completed eight written assignments, four projects, and two exams. All assignments were graded blindly; graders were not aware which students came from the online class and which came from the residential class, and each grader received assignments to grade from both sections. Thus, in terms of duplicating the learning seen in the residential KBAI class in the past, the OMSCS offering of CS7637 appears to be successful: students in the OMSCS section performed as well as or better than students in the residential class. In fact some of the projects submitted by the OMSCS students were of such high quality that they led to a research publication (Joyner et al. 2015).

Possible explanations for the superior performance One explanation for the superior performance of the online section compared to the residential section may lie in the student demographics. On average the students in the online section were older, more educated, and had significantly more programming experience. Again, Goel & Joyner (2015) provide more details about the student demographics in the two sections.

Evolution of the Course

Based on this positive experience, we have taken several steps to promote teaching AI online. First, we taught the

KBAI course again in Spring 2015 and Summer 2015, and are presently teaching it yet again in Fall 2015. Second, we have agreed to increase the size of the individual offerings of the course; while the Fall 2014 section was capped at 200 students and the Spring 2015 was capped at 300, the Summer 2015 section was capped at 400, and the Fall 2015 section has a cap of 250 students. This will bring the course to more than 1000 students through Fall 2015. Third, in Spring 2015, we partnered with Georgia Tech Professional Education (GTPE) to offer a more open section of KBAI: anyone could join the GTPE section, and about 20 students did. Fourth, all KBAI video lessons are now freely available to anyone (www.udacity.com/course/ud409). Teachers at other colleges are welcome to use the materials in whole or part, and anyone in the world can access all video lectures for the course, as well as the microexercises and nanotutors.

Conclusions

This is an exciting time to be teaching and learning about AI. In this paper we presented the design, development, and delivery of an online course on the cognitive systems school of AI titled knowledge-based AI. The cognitive systems paradigm goes back to the earliest days of AI in the 1950s, with the twin goals of using our understanding of human mind to inspire the design of intelligent systems and using our understanding of intelligent systems for insights into the design of human mind (Langley 2012). We can only hope that the availability of this course inspires more students to learn about AI and cognitive systems, inspires other schools to increase their investment in cognitive systems education and research, and inspires other professors and colleges to develop their own online courses on AI..

Acknowledgements

We thank Aaron Gross (Udacity) for taping and editing the video lessons for the online KBAI class, and Joe Gonzales (Georgia Tech) for the Peer Feedback tool.

References

Bringsjord, S., & Schimanski, B. (2003). What is artificial intelligence? Psychometric AI as an answer. In *International Joint Conferences on Artificial Intelligence.* 887-893.

Carpenter, P., Just, M., & Shell, P. (1990). What One Intelligence Test Measures: A Theoretical Account of the Processing in the Raven Progressive Matrices Test. *Psychological Review*, *97*(3), 404.

Connelly, D., & Goel, A. (2013) Paradigms of AI Programming in Python. In Proc. Fourth Symposium on Educational Advances in AI (EAAI-2013), Bellevue, Washington, July 2013.

Goel, A. (1994) Teaching Introductory Artificial Intelligence: A Design Stance. In *Proc. 1994 AAAI Fall Symposium on Improving Introductory Instruction of Artificial Intelligence*, New Orleans, November 1994.

Goel, A., & Davies, J. (2011) Artificial Intelligence. In *Handbook of Intelligence*, R. Sternberg & S. Kauffman (editors), Chapter 23, pp. 468-484, Cambridge University Press, 3rd Edition, 2011.

Goel, A., & Joyner, J. (2015) An Experiment in Teaching Cognitive Systems Online. Georgia Tech School of Interactive Computing Technical Report GT-IC-15-01 (http://hdl.handle.net/1853/53704).

Goel, A., Kun, M., Joyner, D., & Vattam, S. (2013). Learning about Representational Modality: Design and Programming Projects for Knowledge-Based AI. In *Procs. Fourth AAAI Symposium on Educational Advances in Artificial Intelligence*.

Hunt, E. (1974). Quote the raven? Nevermore! In L. W. Gregg (Ed.), *Knowledge and Cognition*. 129-158. Hillsdale, NJ: Erlbaum.

Joyner, D., Bedwell, D., Graham, C., Lemmon, W., Martinez, O, & Goel, A. (2015) Using Human Computation to Acquire Novel Methods for Addressing Visual Analogy Problems on Intelligence Tests. In *Procs. Sixth International Conference on Computational Creativity.* Park City, Utah.

Joyner, D. & Goel, A. (2015) Improving Inquiry-Driven Modeling in Science Education through Interaction with Intelligent Tutoring Agents. In *Proc. 20th ACM Conference on Intelligent User Interfaces*, Atlanta, Georgia, pp. 5-16.

Kunda, M., McGreggor, K., & Goel, A. (2013) A Computational Model for Solving Problems from the Raven's Progressive Matrices Intelligence test using Iconic Visual Representations. *Cognitive Systems Research,* 22:47-66.

Langley P. (2012) The cognitive systems paradigm. *Advances in Cognitive Systems* 1:3–13.

Lehman, J., Laird, J., & Rosenbloom, P. (2006). A Gentle Introduction to Soar, An Architecture for Human Cognition: 2006 Update.

http://ai.eecs.umich.edu/soar/sitemaker/docs/misc/GentleIntroduction-2006.pdf

Lovett, A., Tomai, E., Forbus, K. & Usher, J. (2009). Solving geometric analogy problems through two-stage analogical mapping. *Cognitive Science 33*(7), 1192-1231.

McGreggor, K., Kunda, M., & Goel, A. (2014) Fractal and Ravens. *Artificial Intelligence* 215: 1-23.

Nilsson, N. (1998) *Principles of Artificial Intelligence: A New Synthesis.* Morgan Kauffman.

Norvig, P. (1992) *Paradigms of AI Programming*. Morgan Kauffman.

Raven, J., Raven, J. C., & Court, J. H. (1998). *Manual for Raven's Progressive Matrices and Vocabulary Scales*. San Antonio, TX: Harcourt Assessment.

Russell, S., & Norvig, P. (2009) *Artificial Intelligence: A Modern Approach.* 3rd edition. Prentice Hall.

Stefik, M. (1995) *Knowledge Systems.* Morgan Kauffman.

Winston, P. (1993) *Artificial Intelligence.* 3rd Edition, MIT Press.