Teaching Integrated AI Through Interdisciplinary Project-Driven Courses

Eric Eaton

■ *Different subfields of AI (such as vision, learning, reasoning, and planning) are often studied in isolation, both in individual courses and in the research literature. This promulgates the idea that these different AI capabilities can easily be integrated later, whereas, in practice, developing integrated AI systems remains an open challenge for both research and industry. Interdisciplinary project-driven courses can fill this gap in AI education, providing challenging problems that require the integration of multiple AI methods. This article explores teaching integrated AI through two project-driven courses: a capstone-style graduate course in advanced robotics, and an undergraduate course on computational sustainability and assistive computing. In addition to studying the integration of AI techniques, these courses provide students with practical applications experience and exposure to social issues of AI and computing. My hope is that other instructors find these courses as useful examples for constructing their own project-driven courses to teach integrated AI.*

"It's the sum of many parts which grow to be something great." — *Oliver Bierhoff*

University students often study artificial intelligence (AI) first in a general AI course, which provides a survey of the field, and then through specialized courses on specific AI subfields. This pattern mirrors the fragmentation of AI into subdisciplines that study individual AI capabilities, such as vision, learning, reasoning, planning ,and manipulation, in relative isolation. Over the past several decades, robust research communities have sprung up around each of these topics, providing specialized publications and conferences.

However, from the earliest work on Shakey (SRI International 1969), one of the largest challenges in our field has been the integration of these different AI capabilities into a cohesive system. Despite all scientific progress our field has made in developing methods for vision, learning, reasoning, and others, we have only just begun to develop principled methods for engineering AI systems that integrate these different capabilities (Murphy 2015). Indeed, integrated AI is

Integrated AI for Robotics

… In this project-driven seminar course, students will study and develop an intelligent personal robot assistant, integrating perception, manipulation, learning, planning, and interaction. The resulting versatile service robot will be capable of learning and performing a variety of tasks in real-world environments and collaborating effectively with humans. In addition, students will study a variety of advanced AI topics, including high-level perception and reasoning, scalable knowledge representation, multitask / lifelong learning, integration of perception and control, learning from demonstration, and human-robot interaction…

Figure 1. Integrated AI for Robotics.

frequently cited as a major long-term challenge for AI (Brachman 2006). AI practitioners in industry and research laboratories also often list integrated AI and AI systems engineering as primary learning outcomes that should be included in AI courses (Wollowski et al. 2016).

Interdisciplinary project-driven courses provide a unique opportunity for students to obtain experience developing integrated AI systems, bridging across their experience in more specialized courses. Such a course goes far beyond the end-of-term project often included in standard AI courses, which have limited duration and must balance with the rest of the course curriculum. Project-driven courses give students practical experience in working with a diverse team, negotiating the challenges of deploying a real AI system, and expose them to broader issues of AI, such as its social impact.

This article explores teaching integrated AI through project-driven instruction, focusing on two specific example courses. First, it discusses a graduate course, Integrated AI for Robotics, that is taught at the University of Pennsylvania, involving students with backgrounds in computer science, mechanical engineering, electrical engineering, and systems engineering. After providing an overview of the course, it discusses various pedagogical issues that arise in teaching an interdisciplinary course involving large-scale team-driven projects with students from different backgrounds. The article then shows how these same aspects can be handled in a different way, through the undergraduate course Computational Sustainability and Assistive Computing that was taught at Bryn Mawr College, a small liberal arts college. I hope that instructors may find these two example courses illustrative, and that they may provide mechanisms for creating your own project-driven courses to teach interdisciplinary AI.

Teaching Integrated AI Through Advanced Robotics

At most research universities, graduate students studying AI or robotics choose from a diverse set of courses that cover individual aspects of intelligence. In the case of robotics students, this set is extended to include courses such as mechatronics, control theory, and manipulation. At the University of Pennsylvania, although applications that combined different AI methods were briefly discussed in these courses, there was not a forum for students to study the integration of these different AI functions into a complete intelligent system.

To remedy this issue, I created a project-driven course called Integrated AI for Robotics (figure 1) that is intended for second-year (or later) graduate students. This course combines seminar-style discussions on current research papers with a team-driven semester project in developing intelligent mobile service robots.

Service robotics provides excellent applications for studying integrated AI, since these robots must operate alongside humans for extended time periods in complex, unstructured, uncertain environments, requiring substantial intelligence. Notably, service robots must be capable of handling a wide variety of tasks in everyday domestic or commercial environments, such as finding a lost possession, retrieving office supplies, delivering messages, guiding a tour group, or finding a person in a crowd.[1] To achieve such diverse tasks, versatile service robots must integrate numerous AI capabilities, including perception, planning, reasoning, learning, manipulation, and human-robot interaction.

Team Structure

The multidisciplinary and project-driven nature of the course makes it possible for students with diverse backgrounds to contribute to and benefit from it. For example, some students have extensive experience in mechanical engineering and control, but no experience in traditional AI topics, while others only have experience in statistical machine learning and computer vision. For this reason, the only prerequisite for the course is that students must have taken at least two courses in AI (such as computer vision, machine learning, or general AI) or robotics (mechatronics, manipulation, control theory, and others). The first assignment, which is completed individually and focuses on getting started with the robot operating system (ROS) and the robots, prepares everyone to contribute to robotic software development regardless of their background. Students then complete a short group assignment in which they investigate AI capabilities that are provided by existing ROS software, and program the robots to explore and map an unknown environment autonomously.

Based on a survey of coursework and interest, each student is then assigned initially to 1–2 focus groups, each of which specializes in one AI capability. Example focus groups may include (1) mechatronics, (2) SLAM and navigation, (3) object recognition and scene understanding, (4) manipulation and learning, and/or (5) human-robot interaction. Through class discussions, we determine the AI capabilities that will be useful across multiple service robot projects, and will be developed by each focus group.

In parallel, each student is also assigned to a robot project team to develop a single service robot, focusing on specific service tasks or applications chosen by the team. These teams are best assigned approximately one-third of the way through the semester, after students have gained some experience working with the robots and have had time to generate project ideas. Since each robot team must integrate numerous AI capabilities, the team members are drawn from the diverse focus groups, ensuring balance within each team across the different capabilities and in terms of experience (figure 2). This dual team and focus-group structure resembles the jigsaw educational technique (Aronson et al. 1978), but adapted to a project setting. The focus groups develop software and hardware for each of the individual capabilities, based on the collective needs of the projects, which can then be shared across the project teams. Students are encouraged to use external software and existing ROS packages wherever possible, placing the emphasis on integration rather than the development of the AI capabilities.

As the projects evolve over the course of the semester, we dynamically create new focus groups as the need for new capabilities arise, and retire unnecessary focus groups as capabilities are completed. As a result of this dynamic organization, students end up working in several different groups over course of the semester, giving them broad experience across multiple aspects of service robotics.

Due to the challenging nature of the projects we maintain an atmosphere of *complete collaboration* between robot teams, with the teams encouraged to build off of each other's progress, facilitated by the crosscutting focus groups. The dual team and focus-group structure encourages ad hoc cooperation to develop and integrate the capabilities, while making each student responsible for a specific component of their team's project. Individual project grades are still issued based on a combination of team deliverables and individual contributions to the team and focus group. Students are also required to rigorously cite both external sources and the contributions of their coursemates.

To motivate progress, the course schedule includes concrete deadlines with brief demonstrations every two weeks. In order to keep all teams and focus groups on track, each week includes at least one in-class working session where all teams and focus groups check in with each other and describe their progress, and we work as a group to resolve any issues that arise. We follow the Scrum agile development methodology, where the entire class works in their respective groups to set requirements in the product backlog, organize those items into consecutive two-week sprints for each team and focus group, and hold review and retrospective meetings for planning the next sprint. For coordination and sharing of materials, the groups use a combination of online collaboration sites (such as Slack for project messaging, and Trello for project management), message boards (such as Piazza for course discussions), and code version control systems (such as GitHub).

Service Robot Platform

Current service robot platforms are typically either (relatively) expensive commercial robots (for example, Willow Garage's Personal Robot-2, Rethink Robotics' Baxter, Savioke's Relay) or custom-manufactured research robots (for example, CMU's CoBots, Stanford's STAIR, Boston Dynamic's Atlas), which makes them inaccessible to most educators. In contrast, most robots used in education (such as the TurtleBot 2, iRobot's Create 2, and Adept MobileRobots' Pioneer) are inexpensive, but are little more than mobile bases without the capability of performing multiple service tasks.

Over the past two offerings of the course, we have developed a low-cost service robot platform (Eaton et al. 2016), based on the TurtleBot 2 open-source platform.[2] Our design (figure 3) incorporates a variety of simple and inexpensive modifications to transform the TurtleBot 2 into a 4.5 foot (1.37 meter) mobile indoor service robot, capable of performing a wide variety of service tasks. The modified platform provides a shoulder-height touchscreen and three-dimensional (3D) camera for interaction, significantly upgraded onboard computation, LIDAR for improved localization and navigation, an optional low-cost arm for manipulation, autonomous docking and recharging, and up to 6 hours of run time. Most importantly, the robot can be constructed easily from commercial off-the-shelf components and 3D fabricated parts, making it easy for other educators and researchers to re-create the platform. The design and instructions for the low-cost service robot are available under a free license for education and not-for-profit research.

During the first offering of the course, before we had developed a common hardware platform, each team developed its own custom robot around the same theme on the TurtleBot 2 base. Several of the resulting prototypes are shown in figure 4. Since the diversity of hardware made it more difficult to share

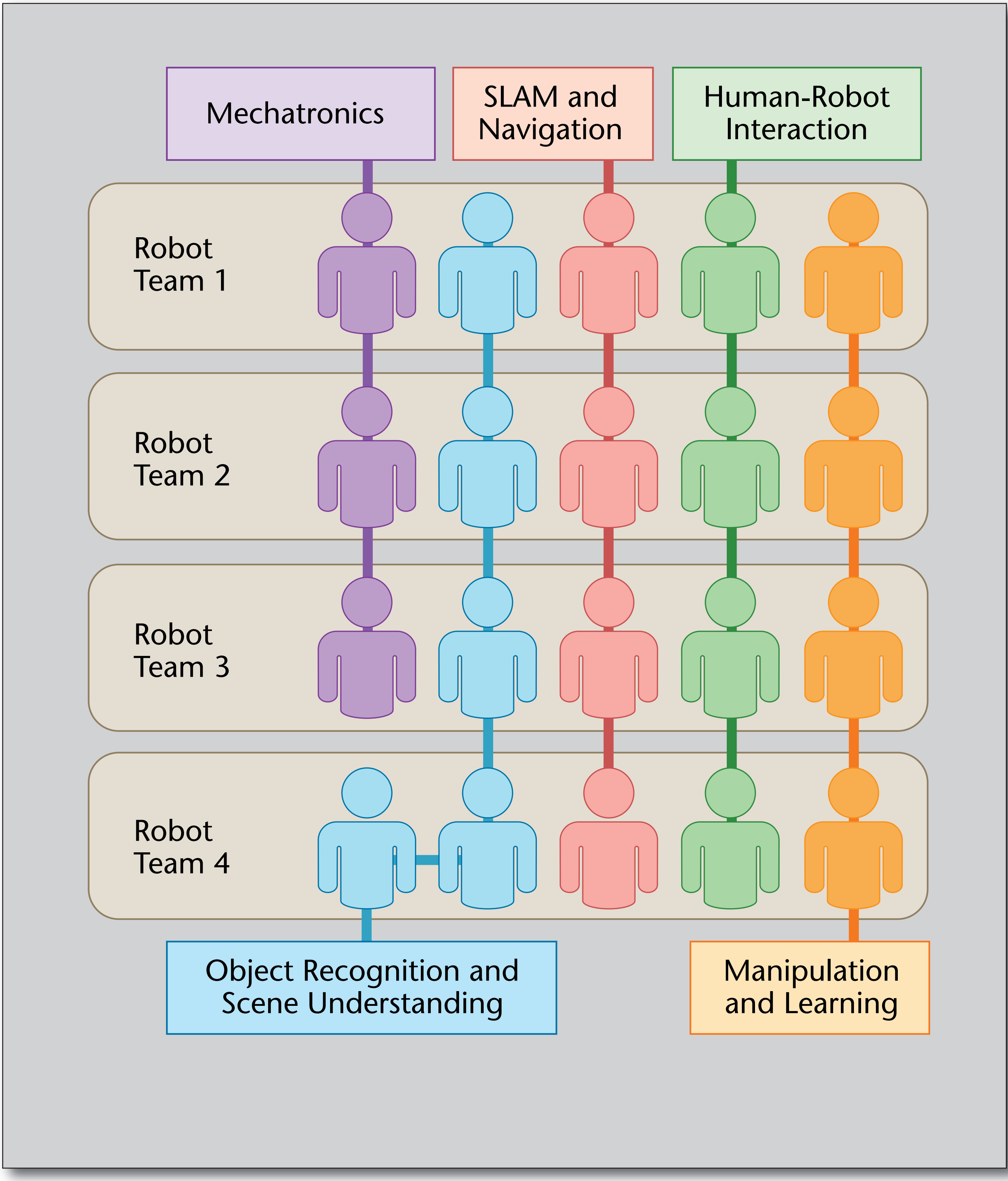

Figure 2. Example Multidisciplinary Robot Teams.

The diagram shows how the topic focus groups crosscut the teams to encourage collaboration. The focus groups concentrate on developing individual AI capabilities, which are then integrated together by the robot teams. This structure also ensures that each student has a specific responsibility to their team's project.

Figure 3. Low-Cost Service Robot Platform.

The platform adds a shoulder-height 3D camera and touchscreen (upper right), upgraded computation (lower right), improved perception, and an arm for manipulation to the widely available TurtleBot 2.

The Robot Operating System (ROS)
Service robotics
Case studies in integrated AI (for example, Stanford STAIR, DARPA urban and robotics challenges, RoboCup)
Architectures for integrated AI
Scalable learning: online learning, transfer and multitask learning
Integrating perception, learning, and control: learning for scene understanding, deep reinforcement learning, perception and manipulation in cluttered environments
Navigation and planning in dynamic environments over long-term deployments
Human-robot interaction: collaborative problem solving, learning from demonstration
Cloud robotics
Evaluation methodologies and experiment design
Social, economic, safety, and privacy considerations, and ethics of personal robotics

Table 1. Research Paper Topics.

Figure 4. Prototype Service Robots.

These service robots were developed by students in CIS 700 at the University of Pennsylvania in fall 2015. These robots were used for a variety of tasks, including waiting tables at a simulated restaurant, object search and retrieval, and voice-driven navigation.

software, we combined the best of each design into the low-cost service robot platform, which will be used by all teams in subsequent offerings of the course.

Seminar Discussions and Course Logistics

In parallel to the semester course project, the students present and discuss research papers on a variety of topics relating to integrated AI. These topics are shown in table 1. In addition, the syllabus includes discussions on project proposals (including Heilmeier's Catechism;[3] see figure 5), project design, and team coordination. Each student's grade is determined based on a combination of individual marks (research paper presentations and summaries, class participation, contributions to the project) and group marks (project proposal, design document, final report and project, and the project website and code repository).

Reflections

Challenging project-driven courses are remarkable in how they bring out students' enthusiasm. Even before the course started, students were especially excited that the project gave them an opportunity to integrate their knowledge from numerous other courses into a single challenging project. This attitude persisted throughout the course, all the way to the final project evaluations. During the final robot demos, the teams eagerly showcased their robots, with all members of the class actively engaged in asking questions and investigating each other's creations. These demonstrations were held in public areas of the university, attracting the interest of passersby.

One of the largest challenges faced during the first offering of this course was simply getting started with such a challenging project. By far, the largest hurdle to getting started was creating the hardware platform. For example, it took several weeks alone to explore different options for the power system for the mobile robot. The final low-cost platform represents a balance between functionality, low cost, ease of recreation, and modularity, aggregating a semester's worth of exploration across four separate robot teams. This common platform allows teams to get busy immediately with developing the functionality of their robot, while allowing them the flexibility to customize the platform's hardware later if they wish.

The second largest challenge was coordination across the different teams. The notion of the explicit focus groups also emerged throughout the first offering of this course, and was made a key component of the course thereafter. The use of focus groups, coupled with the use of the shared hardware platform, allows students to easily share functional developments across teams, placing the emphasis on integration rather than the development of individual capabilities. One of the products required of each team and focus group is a website with detailed instructions and an associated code repository that other teams, and even future students in the course, can build upon, lending longevity to the projects. The use of an agile development methodology (such as Scrum) along with online project management software (for example, Trello or PivotalTracker) is essential to keep all of these different teams and focus groups coordinated, and to ensure continual progress toward completing the projects.

The types of service tasks explored in the project were partly inspired by RoboCup@Home (van Beek et al. 2015) — an international competition where robots perform relatively simple domestic or commercial service tasks in real environments with all their complexities.[4] This project-driven course could easily lead to fielding a RoboCup@Home team, providing students with the opportunity to continue developing service robots.

Finally, although this service robotics course was designed at the graduate level, it could be adapted to the advanced undergraduate level. With some refinement and improved documentation, the robot hardware and software infrastructure developed over the past two semesters could be adapted to provide scaffolding for undergraduate projects in service robotics, either as part of a course in service robotics or as a focused senior capstone experience. Instead of focusing solely on current research papers, the syllabus could be revised to include lectures on service robotics, ROS, various AI and robotics techniques, and architectures for integrating those techniques together. However, graduate students who have taken the course typically find the service robot project intimidating and challenging, and so special care would need to be taken to make the course and project accessible to undergraduates.

Teaching Integrated AI at the Undergraduate Level Through Socially Aware Projects

Although the service robotics project in its current form may be better suited to graduate study, many of the same ideas can be used to teach integrated AI at the undergraduate level through project-driven courses. As an example, this section describes the undergraduate special topics course called Computational Sustainability and Assistive Computing (figure 6) that was taught at Bryn Mawr College in fall 2010.[5] The course focused on the use of computational

Heilmeier's Catechism

What are you trying to do? Articulate your objectives using absolutely no jargon.

How is it done today, and what are the limits of current practice?

What is new in your approach and why do you think it will be successful?

Who cares? If you succeed, what difference will it make? What are the risks?

What are the midterm and final "exams" to check for success?

How much will it cost? How long will it take? *(optional for student projects)*

Heilmeier's Catechism is a set of questions that should be addressed in any research proposal. These questions are credited to George H. Heilmeier, former director of the Defense Advanced Research Projects Agency (DARPA), and former president and CEO of Bellcore (DARPA, 2016).

Figure 5. Heimeier's Catechism.[3]

Sustainability and Assistive Computing

Explore the use of computers and computational methods for positive change, examining both broader impacts on societal development and environmental sustainability, and narrower improvements to individual lives through assistive technologies. We will cover a variety of interdisciplinary topics, including computational allocation of natural resources, monitoring societal-environmental interactions and impacts, ecological modeling, green computing, assistive technologies for people with disabilities, telemedicine, and computers in the developing world.

Figure 6. Computational Sustainability and Assistive Computing Course Description

Computational Sustainability	Assistive Computing
Species distribution modeling*	Intelligent wheelchairs, smart prosthetics, and assistive robotics*
Electronic waste and green technology	
Sensor placement in water distribution networks*	Human-computer interfaces for people with disabilities*
Efficient power and biofuel usage	Assistive technologies and predictors of technology abandonment
Food and farm optimization*	
Telemedicine and medical informatics*	

Table 2. Course Topics.

Those topics marked with asterisks (*) included significant AI components.

methods for positive change at both the societal and individual levels, exploring a variety of topics in computational sustainability (Gomes 2009) and assistive technologies. As in the robotics course, students developed a semester project in small teams, but with a focus on having an impact to society through the requirement to work with an external organization.

Although not strictly a course in integrated AI, many of the topics and the semester team projects involved the integration of multiple AI techniques. The course covered the topics listed in table 2.

To ensure a balanced discussion between computational sustainability and assistive computing throughout the semester, the course schedule interleaved topics from both categories. In addition, the course reviewed the underlying AI methods and discussed project development (proposals, Heilmeier's Catechism, and other topics), presentation skills (including talk and poster design, and elevator pitches), and how to work with external organizations. As in the advanced robotics course, classes were a mix of lectures by the instructor or guest speakers, student presentations of research papers, seminar discussions on the readings, and in-class workshops on the semester projects.

The course project was more open than the service robotics project, but placed a strong emphasis on developing a project with a *tangible impact* to society. Besides encouraging students to explore the social dimensions of their work, this emphasis provided strong motivation to students. Toward this goal, each team was required to work with an external organization on their chosen project. These external organizations were not chosen ahead of time, which allowed students to experience the full challenge of launching and maintaining an external collaboration.

One team developed an ASL-to-Text chat program that would recognize and transcribe a limited subset of American Sign Language (ASL) visual signs into text, integrating techniques from computer vision, machine learning, and accessible interface design. To ensure that the project would meet the needs of the deaf community, the team worked with staff in the ASL Program at the Penn Language Center at the University of Pennsylvania. Another team worked with contacts at the Pennsylvania Game and Fish Commission to develop a set of educational games on the dangers of overfishing, combining mathematical population models with maximum entropy models learned from data. Since each team developed an independent project from scratch, unlike in the robotics course, this course provided no scaffolding for the projects.

The sustainability and assistive computing projects had significantly smaller scope than the robotics projects, but the chance to address real societal and environmental problems sparked the students' enthusiasm. As with many collaborations, the teams found it challenging to maintain their connections to the external organization. They also experienced the difficulty in working with raw data provided by these organizations, with data cleaning and pre-processing becoming a major factor in obtaining good results, as it is in many machine learning applications.

Concluding Remarks

I find project-driven courses to be extremely rewarding, both for the students and as the instructor. In all cases, the largest challenge is helping students to get started quickly in the project, which would seem to require scaffolding and more closed requirements. However, in these types of courses, I believe that one of the worst mistakes an instructor can make is to restrict the project. Instead, leave it open ended and encourage them to *impress* you — let their creativity take over, give them the chance to push the project as far as possible, and see how far they can go. I prefer to specify the *theme* of the project, such as to build a service robot or to address a problem with a tangible impact to society. The theme becomes the seed for their ideas, driving their work.

With these open-ended requirements, each team needs frequent feedback on their projects every one to two weeks, both from the instructor as well as from the other teams. In the robotics course, students responded positively to having deadlines and demonstrations every two weeks to drive their progress. Weekly in-class status reports from all groups and the use of an agile development methodology both encourage progress and invite feedback, as well as allow problems and issues to be addressed early. Often, students become bogged down in minor issues that can consume extraordinary amounts of time. Warn students to watch out for this, and then use these weekly check-ins to detect such problems.

It is also extremely important to have a flexible syllabus. As the projects develop, you will likely need to add or change topics to address specific needs of the projects. My preference is to be upfront with the students from the first class that the course syllabus and schedule will be highly dynamic. They will then expect and accept changes easily, instead of protesting when the schedule is adjusted. Students in project-driven courses should have influence over the course's direction, so request feedback frequently from students on the course schedule. Reading summaries help ensure that students are keeping up to date with the schedule; having them due electronically with a hard deadline in the evening before each class will help guarantee that students are prepared to participate in the seminar discussion.

Project-driven courses are an amazing experience. Consider teaching integrated AI through one of these courses, and enjoy the many benefits for both you and your students.

Notes

1. See A Roadmap for US Robotics: From Internet to Robotics 2013 Edition, Robotics Virtual Organization. (robotics-vo.us/sites/default/files/2013%20Robotics%20Roadmap-rs.pdf)

2. www.turtlebot.com.

3. See the Defense Advanced Research Projects Agency's page on the Heilmeier Catechism, www.darpa.mil/work-with-us/heilmeier-catechism

4. www.robocupathome.org.

5. cs.brynmawr.edu/Courses/cs380/fall2010.

References

Aronson, E.; Blaney, N.; Stephan, C.; Sikes, J.; and Snapp, M. 1978. *The Jigsaw Classroom*. Beverly Hills, CA: Sage Publishing Company.

Brachman, R. J. 2006. (AA)AI More Than the Sum of Its Parts. *AI Magazine* 27(4): 19–34.

Eaton, E.; Mucchiani, C.; Mohan, M.; Isele, D.; Luna, J. M.; and Clingerman, C. 2016. Design of a Low-Cost Platform for Autonomous Mobile Service Robots. Paper presented at the IJCAI-16 Workshop on Autonomous Mobile Service Robots, New York, New York, 11 July.

Gomes, C. 2009. Computational Sustainability: Computational Methods for a Sustainable Environment, Economy, and Society. National Academy of Engineering. *The Bridge on Frontiers of Engineering* 39(4).

Murphy, R. R. 2015. Meta-Analysis of Autonomy at the DARPA Robotics Challenge Trials. *Journal of Field Robotics* 32(2): 189–191. doi.org/10.1002/rob.21578

SRI International. 1969. *SHAKEY: Experimentation in Robot Learning and Planning* (Video). Menlo Park, CA: SRI International.

van Beek, L.; Chen, K.; Holz, D.; Matamoros, M.; Rascon, C.; Rudinac, M.; Ruiz des Solar, J.; and Wachsmuth, S. 2015. RoboCup@Home 2015: Rules and Regulations (online). Palo Alto, CA: Robocup Federation. (www.robocupathome.org/rules/2015_rulebook.pdf.)

Wollowski, M.; Selkowitz, R.; Brown, L. E.; Goel, A.; Luger, G.; Marshall, J.; Neel, A.; Neller, T.; and Norvig, P. 2016. A Survey of Current Practice and Teaching of AI. The Sixth Symposium on Educational Advances in Artificial Intelligence (EAAI-16). In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. Palo Alto, CA: AAAI Press.

Eric Eaton is a faculty member in the Department of Computer and Information Science at the University of Pennsylvania, and a member of the General Robotics, Automation, Sensing, and Perception (GRASP) lab. Prior to joining Penn, he was a visiting assistant professor in the computer science department at Bryn Mawr College, and a senior research scientist at Lockheed Martin Advanced Technology Laboratories. His primary research interests lie in the fields of machine learning, AI, and data mining with applications to service robotics, environmental sustainability, and medicine. In particular, his research focuses on developing lifelong machine-learning systems that learn numerous tasks over a lifetime of experience in complex dynamic environments, transfer learned knowledge to rapidly acquire new abilities, and collaborate effectively with humans and other agents.