

Engineering Informatics for Bio-Inspired Robots

Carie Windham

Edited by Diana G. Oblinger

ELI Paper 10: 2007

October 2007

Abstract

Engineering Informatics for Bio-Inspired Robots is an NSF-funded project trying to break down barriers to collaboration between engineering and computer science departments. Faculty members participating in the project are teaching the next generation of engineers and computer scientists to produce intelligent design by reaching across disciplines. The goal is to ensure that students are better prepared to tackle the multidisciplinary demands of the engineering and computing industries. As part of the project, participating institutions are leveraging cyberinfrastructure to share expertise, store research results, and advocate for multidisciplinary instruction, demonstrating the power of inter-institutional cooperation for teaching and learning.

When Drexel University computer science professor William Regli examines his new Apple iPhone, he doesn't just see a glossy screen and intuitive features. He sees connections: conversations between designers and programmers to ensure the software and screen design work together to produce a fluid touch screen, collaboration between design teams and software engineers to ensure the processor can fit into a sleek design.

"To design that iPhone, you've got to have people with a whole bunch of perspectives in the room providing input on how this thing can look and feel," he says. "That [type of collaboration] is exactly what universities abhor teaching. In the university, things are partitioned in departments."

Regli and his collaborators in Engineering Informatics for Bio-Inspired Robots, an NSF-funded CI-TEAM project,¹ are trying to break down those partitions. Using snake-like search-and-rescue robots and beetle-like Roombas (saucer-shaped robotic vacuum cleaners), they're teaching the next generation of engineers and computer scientists how to produce intelligent design by reaching across disciplines. It's not always an easy task, and the multidisciplinary approach doesn't come naturally to most students. But the struggle, they believe, will ensure that students are better prepared to tackle the multidisciplinary demands of the engineering and computing industries.

"People with backgrounds [in both computer science and engineering] are very rare," says Ming Lin, a computer science professor at the University of North Carolina at Chapel Hill (UNC-CH). "Part of the problem is that our education is traditionally very compartmentalized—we teach our students to think in only a certain way. We don't expose them to the other kinds of engineering problems or scientific challenges that are out there. They can't think outside the box."

Their collaboration is also about teaching faculty members to think outside the box. To pull it off, several institutions are leveraging cyberinfrastructure (CI) to share expertise, store research results, and advocate for multidisciplinary instruction. They're showing themselves, and their colleagues, the power of inter-institutional cooperation for teaching and learning.

What Is It?

Engineering Informatics for Bio-Inspired Robots has roots in more than 50 years of technological change. As software-driven products have become increasingly high tech over time, the curriculum for aspiring engineers and computer scientists hasn't always kept pace. It's a well-articulated problem in publications from the National Academy of Engineering, in which researchers have pointed out that IT has changed product design in artifacts like GPS devices or cell phones. Today's products are information dense, requiring a team that's well-versed in both the mechanics of the product and the underlying IT systems.

To meet the needs of today's complex systems, scholars often turn to informatics, the science of information and information systems. The field is often studied alongside a specific subject, such as biology or the social sciences, to create discipline-specific tools and theories for the management of data. But standard informatics courses don't parse data that's as complex or specific as that required by engineers. To launch an engineering informatics course, it can be difficult to find professors with an extensive background in both engineering and information technology.²

Employers contend that it is nearly impossible to find good hires with expertise in engineering informatics or engineers with expertise in complementary disciplines. Many engineers graduate without a course in programming on their transcripts, and computer scientists often

lack an appreciation for the function of design. “This project came out of the realization that this is a really big problem for the whole country and the whole industry,” Lin says.

With a common vision to teach engineering informatics, researchers at four institutions looked to cyberinfrastructure for a solution. CI (high-performance computing, high-speed networks, grids, massive databases, and advanced computational tools) is as much about people as it is about processing power, data repositories, and visualization techniques. CI allows distributed communities of scholars to work together effectively, creating virtual organizations with the collective ability to solve more complex problems—and solve them more rapidly—than individual researchers working at multiple institutions. If collaborators are unable to find a single professor to teach an extensive course on engineering informatics, they can use CI to link multiple professors together to produce a single course. Furthermore, they could use CI tools to build a repository to store learning modules and research results.

The researchers knew they would need a specific project in which to anchor the curriculum in engineering informatics. Satyandra Gupta, professor of mechanical engineering at the University of Maryland, who was already teaching courses in bio-inspired robot design, suggested robotics, and the other collaborators agreed. Not only would the robots be an intelligent device demanding expertise in computer science and engineering, but, as Regli says, “Kids love robots.” They’re gender-neutral, accessible to people from K–12 through higher education, and appeal to a generation transfixed by Power Rangers and Transformers.

With robotics providing a tangible example of engineering informatics at work, they broke down the curriculum into four components, drawn from their strengths:

- *Design.* What does a bio-inspired robot look like? How do you conceive of these artifacts?
- *Modeling.* How do you transform that design into a workable prototype?
- *Kinematics and Dynamics.* How do you move from a robot to a moveable, programmable machine? How do you program different motions for a snake-like robot? Or instruct another robot to lift its leg to climb a step?
- *Virtual Simulation.* If you cannot physically build the robot, how do you simulate the machine in a 3D environment? How do you test the robot’s function within a simulation?

Using those four components, the collaborators began developing course materials based on their own research and experience, storing the materials in a project wiki. That way, anyone could use slides and lectures developed at another institution. Gupta, for instance, developed video and PowerPoint lectures on bio-inspired robots. Once the project was under way, he used lectures from UNC-CH and Drexel University to integrate more informatics content into his courses. The idea was to give each professor the tools to create his own informatics course by selecting à la carte the content he wasn’t necessarily equipped to teach. “We’re trying to find a way to make it easier to teach people in this new way and get all those perspectives around the table when existing departments can’t afford to have a staff, the way Apple has,” Regli says.

With the online curriculum as a guide, students are taken through each phase of the design process, learning how to conceptualize, produce, and program their robots. Many chart their successes and failures in journal-like postings to the course wiki, storing their robot designs for fellow and future students. These designs—and the collaborators’ course materials—will add to the project’s repository.

Who Uses It?

The project launched with four researchers from four institutions: Regli, Lin, Gupta, and Nicola Ferrier, a mechanical engineer at the University of Wisconsin–Madison. They are each contributing resources to the project wiki and integrating their colleagues' work into their own courses. Gupta and Shapiro teach mechanical engineering students, while Lin and Regli work primarily with computer scientists, although specific classes may have a cross-section of disciplines. The course content has been delivered in upper-level graduate courses and first-year seminar courses for students in engineering and computer science. Though the project has been implemented in higher education, collaborators don't rule out a K–12 implementation in the future.

How Does It Work?

Without collaborative tools, the four researchers were functioning like swimmers trapped in their own discipline-specific swim lanes. Each had an interest in computer science or engineering, but the ropes had to be torn down for the four to move, as a unit, toward engineering informatics.

Their collaboration is seemingly low tech. Having been colleagues for many years, the team uses e-mail, phone calls, and teleconferencing tools to talk about the administrative side of the project. Most of these conversations took place in the beginning, when they were considering what form a portal might take or what specific project they were going to focus on. They also take advantage of academic conferences to arrange opportunities for face-to-face meetings. Regli calls it “pleasantly loosely configured.”

While no one hesitates to pick up the phone to troubleshoot a problem in the classroom or to brainstorm the next step, the actual fruits of their collaboration (lesson plans, case studies, 3D models) are stored in a project wiki hosted on a Drexel University server. The wiki is one part archive, one part toolbox, and one part digital journal. In many ways, it's a virtual intersection for the various components of the project.

Users can click on an individual course title—such as Regli's fall course in Bio-Inspired Robot Design—and navigate toward a course wiki page. The page—which is effectively a digital, editable syllabus—offers a synopsis of the course, Web links, and a detailed course schedule with links to individual lesson PowerPoint presentations, which, though contained on a single course site, are contributed by professors at different institutions. The presentations can be freely downloaded by students in the course or shown by lecturers in the classroom. For outsiders, it's a how-to guide to conducting a course in robot design.

Users can also use the wiki to read about each student's robot design. A page maintained by student Shawn Pierce offers a week-by-week breakdown of his journey to create the “Maticore,” a “fearsome beast which has the body of a lion, the tail of a scorpion, and the wings of a bat.” Pierce includes a cartoon-like rendering of a “real” Maticore to complement his blue, yellow, and gray Lego version. With each weekly progress update, he includes pictures of the robot's evolution and, eventually, 3D virtual models. Beyond its role as repository and showcase, the wiki also serves as a portal to the Web, giving students and researchers links to videos on robotics (contributed by Gupta) and sites on robot design and theory.

The wiki's main purpose is as a resource for each faculty member's classes. Students in Regli's seminar might be asked to navigate a Roomba through a maze, while Gupta's graduate students might be forcing a snake-like robot to move across the floor. Each class

has access to the wiki's resources, and each class can contribute to the database. This two-way street—students and collaborators leveraging the wiki's resources while contributing their own—fuels the project's content and longevity.

Why Does It Work?

After teaching 650 first-year engineering students how to coax their Roomba vacuums through mazes, Regli holds no illusions about the popularity of the course. “For about 40 percent [of the class], it was great. It was fun. It took a lot of time, but it was very rewarding,” Regli says. “The other 60 percent would say, ‘I’m a [civil or structural] engineer. Why do I have to learn about robots?’ They didn’t see it as a microcosm of what they might be dealing with as engineers.”

However, Regli doesn't think those numbers necessarily mean the project isn't working; he has no doubt that many years down the road, his students will look back on that freshman-year course and remember the value of multidisciplinary teams and collaboration. “As a professor, I know these students are going to remember this assignment for years to come,” he says.

Students remember because the courses revolve around a complex, hands-on problem. Without a clear path to follow and innumerable obstacles along the way (mistakes in coding, misplaced parts, or other unexplained glitches), students are forced to engage in critical thinking and outside research to finish the task. The element of frustration in the process carves a deep impression in their minds, putting the emphasis on *how* they approached the task, not whether they finished it. For students who complete it, the sense of accomplishment and the real, tangible product at the end of the semester give them an example of their work to show potential employers or admissions officers, particularly if they choose to chronicle their efforts on the course wiki.

On a broader level, the collaboration benefits from the multi-institutional approach. The collaborators chose to work across four institutions instead of four departments because they wanted to widen the project's impact and because they wanted to create a project that would be immune to a particular institution's culture or bias. The cross-institutional nature of the project demands that all course materials use general principles and tools, not those developed and employed by a single department. This also ensures that future collaborators can have a seamless integration to the project.

The project works, in part, because of the strength of the collaborators' relationships with one another, forged over many years as friends and colleagues. Each participant is a leader in his field, with experience in other disciplines: computer scientists who know some basic engineering, and engineers versed in programming. This ensures that each professor can easily adopt and employ pieces of the curriculum developed by other collaborators.

What Is the Value?

Gupta wasn't waiting for this particular collaboration to start teaching about bio-inspired robots. He's been developing these classes for years, and each time, students clamor to sign up. There's just something about building their own robots that gets students running to his classes. By the end of the semester, however, they're often running the other way.

“The thing that I truly enjoy is that a week before the project is due and they are lost in design and it's never going to work out, they so much want to throw this robot out the window,” Gupta says. They beg, “Can't you just give me a C grade?” Then, when the robot starts

Engineering Informatics

working, the students are practically dancing with glee. “They believe they can take on the world now because they’ve made this little robot walk,” he says. Gupta says it’s in those moments of frustration, not elation, that he sees the real value in their work. “We give students exposure to how the real world is going to be,” he says. “The product is challenging enough that it forces them to use advanced informatics. They get to see a truly interdisciplinary product.”

More than just inspiring students to dance, however, this collaborative project creates value for the field, the institutions, the faculty, and students.

For the Field

Though the term *informatics* has been used for the past half century, less attention has been paid to the field of engineering informatics, a cross-section between traditional engineering disciplines and the computing processes that lead to a higher rate of discovery and design sophistication. In Regli’s iPhone example, engineering informatics is the collaboration between product designers creating a touch screen and computer programmers designing the software capability. It includes a wide range of computing tools at the engineer’s disposal, from 3D models to large-scale databases.

The project creates a model for studying engineering informatics from a practical perspective, creating learning modules that can be used at other institutions. It also plays a role in exposing students and departments to the field by presenting students with advanced problems that demand multidisciplinary answers.

The use of a public wiki gives life to the project beyond the careers of the individual collaborators and opens the door to future collaborations. Lin says they like to believe they are “leading the train” in many ways by encouraging faculty and researchers to freely exchange educational materials, not for personal gain but for the future of the field.

For Institutions

One of the many roadblocks to the comprehensive study of engineering informatics is that for many years individual components of the field were taught in silos. An engineer would learn the mechanics behind a device but might never take a course in 3D modeling. A computer scientist might never deal with a data set as complex as one required by structural engineers. The project gives institutions an opportunity to teach content-rich courses without hiring additional faculty. They can leverage the expertise available at other institutions.

For the Faculty

When a new field emerges, it can be difficult for an individual instructor to produce timely, content-specific course material while juggling her own research and teaching demands, particularly if that material requires taking a crash course in a topic outside her field of expertise. The collaboration eliminates that problem by allowing multiple researchers to contribute to a single project, creating a model for faster content evolution. Individual instructors need not step outside their domain to learn the complexities of other sciences. Instead, they can absorb material from other contributors who possess greater expertise, giving students access to experts in the discipline without ever leaving campus. It also gives an individual professor more resources and inspiration for structuring courses, lectures, and suggesting readings.

By capturing and publishing student designs in the wiki, faculty can also give students access to ready-made case studies, allowing them to learn from one another. Moreover, participating researchers benefit from the exposure to other disciplines. By engaging in the collaboration and sharing materials, they learn more about fields outside their discipline.

For the Students

Put simply, students say the classes in robot design are “just fun.” Piecing together Legos, connecting wires, and soldering parts makes for an exciting, hands-on experience. But the fun exterior masks deeper teaching and learning outcomes. Students are forced to work in multidisciplinary teams on complex mechanical and programming projects. An engineer might need to ask a computer programmer for code advice. A programmer might be baffled as to how a single slipped wire can impact performance. Students are forced to rely on each other’s strengths and to teach themselves new skills to get their robots to work.

It all mimics industry trends, collaborators say. The students are faced with the reality that engineers and computer scientists can’t swim in their own lanes. For today’s technological devices to function, their designers must learn how to collaborate and tackle each task with an interdisciplinary approach. “We’re making sure our students are still competitive,” Ferrier says, noting that the movement of many lower-level science and engineering jobs offshore means that American students must be prepared to step into the high-level jobs that require a new level of collaboration. “The jobs that are now available to Americans are changing all the time,” she says, “and so our definition of preparedness has to change.”

What Is Required?

The project’s biggest struggle, according to Gupta, is resources. “Robots aren’t cheap,” he says, adding that arranging resources often consumes most of his time. Some of those supplies are covered from a one-time CI-TEAM grant from the NSF, with individual departments picking up some of the costs. Even with their help, students still pay part of the tab, especially if they want to power their robot with more expensive parts. Parts can cost \$200–\$300 per robot for components including motors, gears, and microprocessors. That doesn’t include equipment costs when students need access to an industry-strength machine or tool for construction. Even if the department agrees to let the student use university resources without charge, they still have to convince researchers to give up valuable time so the class can log time at the machines.

Physical resources are just one piece of the puzzle. Many of the project’s needs are less tangible.

Cooperation

Like any collaboration, the project requires cooperation and resource sharing between institutions and researchers, not always an easy feat in the competitive academic environment. Faculty must be willing to share some of their intellectual property to support the advancement of the entire project.

Infrastructure

The project relies on wiki technology to create the portal for teaching tools, lesson plans, and designs. The wiki allows individual, authenticated users to edit and manage content.

Expertise

The collaboration relies upon expertise delivered by scholars who are leaders in their field yet familiar with the tools and principles of other fields.

Collaborative Tools

To maintain communication, collaborators rely on e-mail, telephones, and teleconferencing.

Where Is It Going?

With the collaborative framework in place, the team hopes to continue building their educational repository so that in the immediate future they can move from integrating each other's materials into their own courses to delivering one unified course in engineering informatics. This way, Gupta says, a student in Maryland would learn the same material as a student at Drexel. There is still some uncertainty about how that course might be delivered—the collaborators have previously experimented with video lectures and exchanged PowerPoint presentations in the past—but the idea is to capture the content and deliver it through the wiki so that they, and other institutions, will have ready access for teaching. The emphasis will be on “plug and play” materials—tools and content modules that can be easily integrated into another professor's class without format or translation fears.

Moving forward hinges on the project's ability to attract more participants, either as contributors or users of the content. “At most, I'm going to teach one class each year,” Regli says. “But if there are 10 or 20 [collaborators], we have a real chance to build a virtual textbook.” As more institutions sign on, Regli sees the project moving past a single module and into a cohesive program offered as a certificate program or a minor in engineering informatics. He'd also like to see it trickle down to K–12 education, using simplified versions of the Lego-block projects currently used in higher education.

“The immediate value is getting to teach some cutting-edge topics in novel ways at our own institutions,” Regli says. “Going forward, because we've done it as a proof of concept, other people will find these resources helpful to them, and the infrastructure will live on. Over time, it's going to become other people doing it.”

Resources

- The Project Wiki Page, [http://gicl.cs.drexel.edu/wiki/CI-TEAMS: Engineering Informatics for Bio-Inspired Robotics](http://gicl.cs.drexel.edu/wiki/CI-TEAMS:EngineeringInformaticsforBio-InspiredRobotics): Navigate the site to learn more about individual lessons or students designs.
- NSF CI-TEAM, <http://www.nsf.gov/crssprgm/ci-team/>: Provides an overview of the NSF's commitment to CI research, including current news and studies on the possibilities of CI for teaching and learning.

Endnotes

1. The NSF CI-TEAM project funds initiatives that introduce new and nontraditional audiences to the fundamentals of cyberinfrastructure. With a mission to prepare future scientists and engineers to use cyber-enabled tools, the project targets proposals that either teach audiences how to use cyberinfrastructure or engage cyberinfrastructure to meet their research and teaching needs.
2. To read more, see William Regli, “The Need for a Science of Engineering Informatics,” *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 21 (2007), http://journals.cambridge.org/download.php?file=%2FAIE%2FAIE21_01%2FS0890060407070102a.pdf&code=bbe812fa864e061a4579326df0a7aa58.