Using an Integrated Platform for Learning™ to Reinvent Engineering Education

Roger L. Traylor, Donald Heer, and Terri S. Fiez, *Senior Member, IEEE*

***Abstract*—The most pressing and critical needs for engineering graduates in 2013 and beyond are to be natural innovators who are able to integrate their knowledge to solve complex engineering problems. This paper introduces an integrated platform for learning™ as a solution to meet these needs. The platform for learning provides an environment for innovation, while integrating a curriculum into a coherent whole.**

***Index Terms*—Design, education, educational technology, electrical engineering, engineering curriculum, innovation, robots, TekBots.**

I. INTRODUCTION

THE CURRENT half-life of engineering knowledge—the time in which half of what an engineer knows becomes obsolete—is estimated to be in the range of 2.5–7.5 years [1]. Generally, the fundamentals engineers learn in school remain fundamental, but the way in which the fundamentals are applied changes rapidly. Without instilling the need to continuously reeducate themselves, adapt to changing conditions, integrate knowledge from various disciplines, and then apply this knowledge in innovative ways, future engineers will find themselves sidelined and, as a consequence, so will the companies that rely on them to drive the technology development.

The education of engineers in the next decade will have a direct impact on how companies compete now and in the future in the global economy. In the new economy, technological innovation is central to wealth creation and economic growth [2]. To sustain a competitive advantage, engineers must be enablers to "wealth creation" rather than simply be a commodity on the global market [2]. As one National Science Foundation leader expressed it:

> "Engineers must be enabled to grasp the opportunities for innovation rather than simply contribute to enhancing productivity. Innovation, especially through engineering enterprise, is at the core of a healthy economy."

A key ingredient of innovation is the ability to design complex systems. Engineers skilled in design must be technologically literate, prepared to readily capitalize on new knowledge and able to effectively utilize contemporary tools and methods for designing new systems [3]. They must understand the entire life cycle of products, from conception to development to deployment through end of life. They must create not only a system that works, but also the right system for the application. A design engineer with a more holistic view of the world is required with the ability to develop innovative solutions and apply design skills across disciplines.

Next-generation engineering problems will span multiple complex systems, such as bioengineering, software engineering, and mechatronics. They will also address global issues, including sustainability, life-cycle engineering, infrastructure, and systems renewal [2]. To solve these multidisciplinary problems, future engineers must have the ability to integrate their knowledge, making connections between topics across different subjects and disciplines. Being able to integrate their knowledge, however, does not mean that depth of knowledge is sacrificed. Deep discipline-specific knowledge is even more tantamount to successful future engineers with the caveat that specific knowledge must be made adaptable to multidisciplinary problems as part of multidisciplinary teams.

The depth and breadth of knowledge and professionalism are at the core of any engineering program [3]. In addition, the complexity of next-generation engineering systems in the new economy requires that engineers of the future have a clear understanding of how engineering concepts are interrelated, and they must be able to leverage this understanding to develop technology innovations. Demonstrating knowledge integration requires comprehensive depth of knowledge and many of the characteristics associated with professionalism, including effective teaming, communication, and understanding of the broad context of engineering.

Many enhancements have been developed over the years to reform engineering education. A number of these approaches address the need to increase student involvement, thereby increasing retention in engineering and the students' overall learning [3]–[5]. Table I summarizes many of these approaches and shows what the impact has been on engineering education.

Active/cooperative learning techniques allow students to take responsibility for their learning and play a key role in ensuring they learn the material in a comprehensive way by being both a learner and a mentor. These techniques are effective to enhance learning as well as retention. By combining these techniques with technology enhancements, students are released from low-level redundant tasks, and they can focus their efforts on understanding complex physical phenomenon and inter-relationships of concepts.

Another commonly used educational strategy is just-in-time learning. Instead of teaching topics in isolation, the inter-relationships of topics are illustrated by presenting the necessary

---

Manuscript received November 6, 2002; revised August 3, 2003. This work was supported in part by the National Science Foundation under Grant EEC-0230679 and by the Tektronix Corporation.

The authors are with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331 USA.

Digital Object Identifier 10.1109/TE.2003.818749

TABLE I
EDUCATION STRATEGIES USED TO ENHANCE ENGINEERING EDUCATION

Educational Strategy	**Description**	**Impact**
Active/cooperative learning [6-15]	Instructional activities engage students in doing and thinking instead of passive listening.	Improved retention. Higher academic achievement. Improved individual accountability. Improved small-group skills. Enhanced creative thinking.
Technology enhancement [11,13,16]	Computing resources introduced into classroom to enhance learning by using software tools.	Increased comfort level using computers as tools. Mundane tasks reduced to allow focus on higher-order thinking.
Just-in-time learning [4,12,17-19]	Theoretical concepts introduced when students' experiences create a demand for them.	Improved academic performance. Life long learning skill development. Theory and practice kept in context.
Curriculum integration [20-25]	Learning activities restructured to build contextual connections between topics.	Enhanced ability to transfer knowledge to new situations. Better program retention because of material relevance. Better recall of material.

background material as it is needed. This task can be particularly challenging for instructors since students do not necessarily have the "traditional" background before they enter discipline-specific courses.

Each of these strategies enhances the student experience in engineering resulting in increased retention in engineering, as well as enhanced learning. There will be a continuing need for these approaches, and/or derivatives of them, to educate the next generation of engineers. However, more advances in educational strategies will be needed specifically to address engineering knowledge integration and innovation [2], [27].

While innovative educational strategies help students understand the material, a major drawback of most engineering curricula is that courses are taught as individual topics in isolated, disconnected pieces. Courses are separated like islands, implying by silence that each topic has little relation to any other and that they are only in some yet undisclosed way related to the practice of engineering. The "big picture" of a particular discipline and how the constituent pieces relate to each other is often not assimilated by the student until late in their education, if ever.

This method of teaching engineering ignores the need for connection and for integration, a technique that should be at the core of an engineering education [2]. Recently, several programs focused their efforts on *curriculum integration* to build connections and relevance between topics [20], [21], [24], [25]. Unlike typical engineering programs where topics are conveniently partitioned into separate classes, this structure highlights the relationships between topics. This approach helps students to learn more efficiently and to apply their knowledge to different situations more effectively. In effect, curriculum integration reduces compartmentalized learning where interrelated topics become disassociated, and it emphasizes the relationships between topics and concepts [17], [26].

One very successful program integrates the first two years of engineering education [20], [21], [24], [25]. Classes in math and physics are coordinated with introductory engineering offerings to enhance the connections between math, science, and engineering courses and to promote communities of learners [22], [24]. This excellent first step represents the kinds of changes needed for next-generation engineering curricula.

In the next section, a platform for learning is defined. This approach is not in opposition or in competition with these aforementioned efforts; rather, it is complementary to and further enhances their impact. Section III describes the attributes of a platform for learning. In Section IV, a detailed description is given of a computer engineering curriculum based on a platform for learning. Finally, Section V concludes by summarizing the engineering skills needed in 2013.

II. DEFINING A PLATFORM FOR LEARNING

A platform for learning is a common unifying object or experience that weaves together the various classes in a curriculum. By employing this common platform in many classes, the interrelationships and interdependencies of the classes are clearly illustrated. A key attribute of the platform is that it becomes a foundation for learning that is built upon as the student progresses through the curriculum. The platform represents what the student has learned, how it is applied, and how it relates to other subjects he or she has learned in other classes.

The preferred learning platform is *hands-on*, and preferably, uses a physical object. The body of research shows that using physical "manipulatives" enhances learning [27]. In addition, the amount of material retained and the ability to integrate that knowledge is greatly improved when the course material relates to personal experience [27]. A platform for learning can take many different forms. For an electrical engineering student, a platform for learning may be a very simplistic mobile base that illustrates the operation and construction of basic electronics. As the student progresses through the curriculum, capability, and functionality is added, such as sensors and a microprocessor that creates a platform for learning with wireless communications, networking, digital signal processing, and other topics all contained on the platform. For a computer science student, a single-chip "computer" with no predefined *brains* may serve as a learning platform. As the computer science student progresses through the curriculum, he or she builds up the brains in the computer.

"Hands-on" does not necessarily imply a physically tangible platform. The natural platform for other disciplines may be virtual or nonphysical. A business student may use a business plan as a platform. The point is that there must be an intensive interaction between student and platform so that the student forms feelings of personal ownership toward the platform.

Much of the research on human learning indicates that theoretical concepts are far easier to learn when the students already have experience with the real-world objects that are the object of the theory [27]. Without a concrete object with which to relate a new abstract concept, learning is often difficult. Using the platform for learning allows new concepts to be introduced in the preexisting context of the platform that the students have worked with before, helping the students assimilate new information based on what is already understood.

The platform should create an environment that closely emulates engineering practice and experience. Learning that is situated in real work within a community of learners supports the development of students' personal identities as capable and confident learners and retainers of knowledge [28], [29]. The platform is a real project that helps create the atmosphere of a large engineering team. The team community sets the stage for the students to engage in formulating and evaluating questions, problems, conjectures, arguments, and explanations, just as professional engineers do in the workplace.

If the platform is a real project, not a "pretend" paper project, it brings with it all the real-world constraints and problems and general "untidiness" of projects found in contemporary engineering. Thus, the procedures, practices, and approaches used by everyday engineers are naturally brought into the classroom or laboratory. This concept is in stark contrast to the often boring, contrived, and sanitized exercises found at the end of the chapter, which have no loose ends or real constraints and typically have only one correct solution.

Real projects have real bugs. However, in preparing students to work on projects in the real world, the instructors teach as if once the design is finished, the job is done, when in fact, it has only begun. The topic of how to detect, find, and fix "bugs" is almost totally neglected in today's engineering curriculum. A good learning platform can go a long way in correcting this shortcoming by thorough immersion in the art of debugging.

Many educational initiatives have included what appears to be a platform for learning. Under the banner of educational reform, some have introduced robots or some other "project" into their curriculum. However, there is a fundamental difference between an integrated platform for learning and a project inserted into a curriculum. The difference is a question of order. The difference is whether the class is adapted to the project or whether the project is adapted to the class and whether the platform is the subject or the object of the subject.

A platform for learning is never applied to or forced onto a class or curriculum. Before a platform is selected, the first question is, "What am I trying to teach?" This question is followed by, "Can I assist the learning process with a platform for learning?" Only when the instructor knows what is to be taught can a proper platform selection be made.

Thus, the platform for learning makes no attempt to be all things to all courses within a curriculum. The platform should be intimately connected to the curriculum's core courses. However, the platform performs more of an adjunct role for courses, such as calculus, chemistry, and physics, where it becomes a motivator for the topic and it establishes relevance by providing a point of reference. For example, a battery-powered robot can motivate and establish relevance for a study of chemical reactions. The study of derivatives can be more illuminating when motivated in the context of a moving and accelerating robot. In this way, a single evolving platform can support related subjects while being closely linked to core subjects.

A platform of learning can also promote the spontaneous creation of new platforms by students themselves. The given platform provides a model for knowledge transfer so that the student creates his or her own platform for "doing something" that utilizes knowledge gained in a class. This outcome is extremely powerful and desirable in using a platform for learning. In this manner, multiple platforms/subplatforms may be spawned that cover peripheral subjects.

III. ATTRIBUTES OF A PLATFORM FOR LEARNING

A learning platform is a thread that runs through a degree program. By using a common platform throughout a degree program, the integration of knowledge is enhanced. The platform provides the conceptual "glue" between lecture topics. When using the platform, lecture topics are related to or discussed in the context of the platform. If all the topics relate to the platform, they will also relate to each other. This statement holds true whether the context is one class or the entire curriculum. Interaction between topics becomes clear when viewed at the point of the platform.

Fig. 1 illustrates the inter-relationship of courses and topics using a platform for learning. Thevenin's theorem, electromagnetic waves, and microcontrollers are three very different topics, taught in different classes and at different times. Suppose a wireless battery-operated robot with on-board microcontroller was the platform of choice. A model of the microcontroller output drivers could be created with a Thevenin's equivalent circuit. The equivalent circuit is connected to a mismatched transmission line that connects to another on-board device. The electromagnetic interference created by "ringing" on the transmission line affects the wireless control receiver. The relevance and inter-relationship between these topics can be clearly and powerfully brought out through the platform. Furthermore, the platform allows discussion of the topics in any of the classes in a way that it relates to the others.

A platform for learning complements the existing structure of lecture and laboratory. It motivates lecture topics and meshes them with laboratory experiences. It acts to expand and integrate the entire curriculum, as illustrated in Fig. 2. Class lectures are effective in providing depth and breadth in the discipline. This accomplishment is complemented by the laboratory, where hands-on experiences reinforce the lecture material. Integrating a platform for learning into the curriculum expands

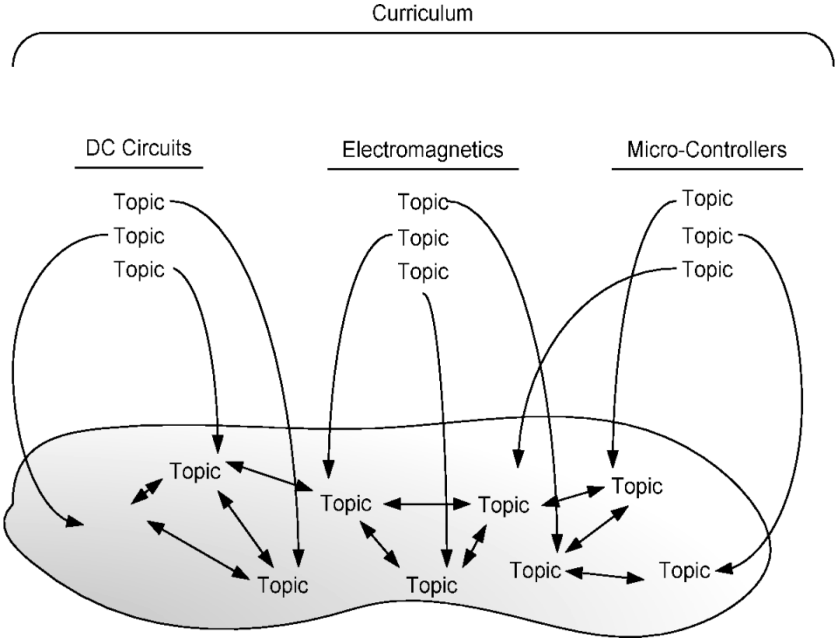

Fig. 1. Courses and topics integrated by a platform for learning.

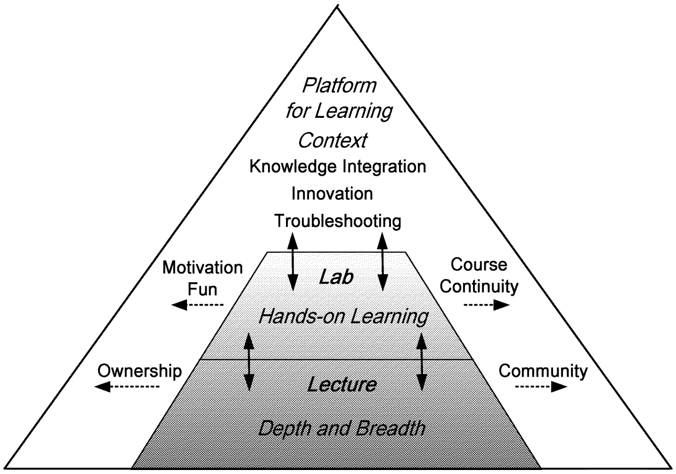

Fig. 2. Platform for learning expands the learning opportunities by providing context, knowledge integration, innovation, and troubleshooting experiences. It also enhances ownership, motivation, community, and course continuity.

the learning opportunities and effectiveness in multiple dimensions. The platform provides a context for learning that allows for connecting the knowledge among classes, developing innovative skills, and enhancing troubleshooting skills.

Using a platform for learning throughout the curriculum provides all the students with a common point of interest, which establishes all students as part of a learning community. A common platform used throughout the program provides a bridge between students at different stages of their studies. These bridges help build a large community of learners who share a common focus, which facilitates learning [28], [29].

Using the same platform for four years does not mean that a fixed platform is used for four years. The platform must undergo evolutionary changes to track the students' abilities and to adapt to the different subject material. In addition, other subplatforms may evolve from the main platform. At the end of a four-year program, the highly sophisticated platform/subplatforms are an embodiment of the students' knowledge.

The learning platform should be personally owned. As students progress through a curriculum, they will have made a considerable investment of time and money. As with any personal item acquired at considerable price, there is a vested interest in the condition of that object. Students may also want to maintain whatever status they have attained with their platform because it is visible and tangible evidence of their ability. To maintain or improve the status of their robot, and thus themselves, they will be motivated to learn and learn more than their peers [30].

The platform must be *flexible* so that students may easily experiment and try out new ideas. Innovation occurs when something new is created. A flexible platform encourages innovation by making new creations easier to make. The flexibility of the platform should make experimentation easy to the point of enticement. If possible, the natural curiosity in students should be stimulated. Experimentation and exploration should be drawn out of the students. This stimulation leads to learning through discovery. The discovery learning process is a gratifying one that helps generate the desire for life-long learning.

A flexible platform is one in which students are not "steered" in their approach toward the problem, nor are they influenced by knowledge of a "correct" solution set. For example, many classes in digital logic design use prototyping boards containing field programmable gate arrays (FPGAs). The boards seem to have every imaginable support part on the board wired to the FPGA and to each other with the hope that anything that anyone would ever want to build would be realizable. What happens, however, is that the solution set becomes fixed by the prewired parts already on the board, and the approach to be taken by default will include the parts that are already on the board. This situation stifles innovation and gives support to the false notion that the outcome of design falls in a fixed solution set.

Flexibility must be balanced with the ease of use. Building blocks of the platform must be understood by the students and be easy to form into something new. If the first step of creating something new is too hard, the students may be discouraged from ever trying. Therefore the platform needs to be matched to the students' capability and be able to change with them as they progress through the learning process.

Flexibility also means that students can access the platform in a way convenient to them. When curiosity strikes, the platform should be accessible. If the learning platform is located in a locked laboratory, the opportunity for experimentation and innovation is lost. The optimum situation is one in which the platform is accessible at any time and in any place.

IV. A CURRICULUM BASED ON AN INTEGRATED PLATFORM FOR LEARNING

In this section, the authors illustrate a curriculum for computer engineering based on a robot platform for learning, referred to as a *TekBot*™. The basic platform is described, followed by how it is used to reinforce the lecture and laboratory materials and integrate the topics from different classes. This platform is particularly well suited to reinforcing the link between hardware and software and enhancing innovation skills while fostering community among the students.

The students begin the freshman year in an introductory electrical and computer engineering class where fundamentals are presented in lecture and then brought to life as students apply these fundamentals to construct their individual platform or robot. As students progress through their four-year program, they are exposed to more complex theoretical principles and add new capabilities to their TekBot. This progression connects the topics from one course to another and provides opportunities for putting the theory into practice.

The Tekbot platform for learning begins as a small motorized robot that has two touch-sensitive switches that detect contact with a wall or vertical surface. If either switch makes contact with a wall, the robot backs up, turns away from the wall, and then proceeds forward again. It is powered by six NiCd cells and has a battery charging circuitry on board. Modified servomotors drive two foam wheels. Above the aluminum chassis is a Plexiglas plate that is the mounting base for any circuitry.

The assembled basic platform is shown in Fig. 3(a). Students are initially given a bag of parts that they use to create their TekBot. The TekBot includes an analog controller board, a motor controller board, and a prototyping board. This modular construction allows for boards to be added and removed as the students progress through the curriculum.

The Tekbot platform is designed around three criteria. First, the platform must enhance the concepts presented in lecture and laboratory. A concerted effort to minimize the parts count would make the platform less expensive but would diminish its educational value. For example, an encapsulated H-bridge integrated circuit would simplify the design and reduce the overall cost. By fully exposing all the transistors and their biasing with a discrete H-bridge, however, the platform is a far richer teaching tool. This board is shown on the right of Fig. 3(a).

The second requirement of the platform is that it is flexible. The platform should be constructed so that experimentation is as unrestricted as possible. This goal is achieved by making the basic platform as simple and unencumbered as possible and using commonly available parts. For example, students may make a four-wheel-drive version by mounting two aluminum bases back to back and adding two extra motors. All students are encouraged to modify and improve their platforms within the constraints of being able to complete the laboratory exercises.

Care is taken not to supply everything the students might need because this provision stifles innovation and hinders imaginative thinking. However, provision is made for experimentation, including a prototyping board and a prototyping area on one of the circuit boards. The intention is to leave open as many avenues for experimentation as possible.

The final criterion is that the platform should be reasonably priced so that all students can own their individual platform. Personal ownership of the platform is considered a necessity. The primary effect of personal ownership is that it creates a sense of ownership and pride, enhancing the desire for learning. Allowing students to experiment with their platforms at any time and in any place promotes life-long learning.

Innovation in the curriculum is encouraged by including challenge problems with each laboratory. These challenge problems are design problems with many solutions. They typically draw off the expertise the students have acquired up to that point and stretch their skills and imagination. Most often, successfully completing the challenge problems means the students have a more capable TekBot.

The issue of personal ownership brings out one of the distinctive characteristics of this curriculum using a platform for learning. In this curriculum, the "laboratories go with the students." The students own the objects of the laboratory plus the tools required to work in the laboratory. These *tools* include both the custom hardware and the off-the-shelf software. In this environment, "laboratories" can happen anywhere, any time.

Having obtained an understanding of the physical platform, it is now beneficial to describe the way the platform serves to interconnect the curriculum. The overall core computer engineering curriculum (on a quarter system) is shown in Fig. 4, with a summary of course topics given in Table II. The supporting mathematics, humanities, social sciences, and other general requirements are not included so that the key features of this curriculum can be highlighted. What makes this curriculum particularly distinctive is the tight coupling and inter-relationships of the various courses (as shown by the interconnections), the opportunities for student innovation, and the hardware/software co-design throughout the four-year curriculum. All three of these features are reinforced with the TekBot platform for learning.

Fig. 4 shows four core thrusts in the first three years: electronics, computer hardware, signal processing, and computer software. Traditionally, these are distinct and separate classes and topics, but by using the platform for learning, the topics become interdependent. Once this core curriculum is completed, students can specialize by selecting electives in specialty areas, including but not limited to, communication systems, embedded systems, VLSI system design, biomedical

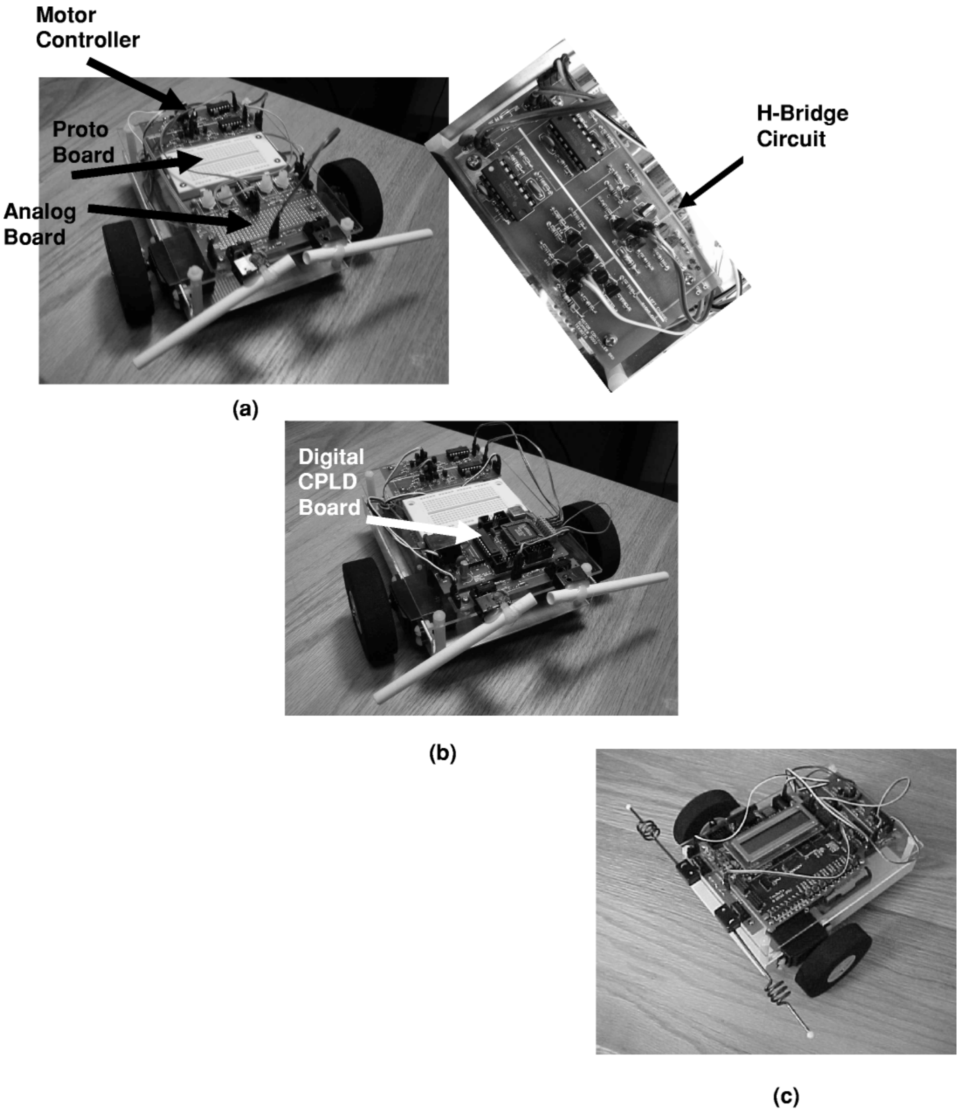

Fig. 3. TekBot platform for learning. (a) Basic analog platform. (b) Enhanced digital platform. (c) Microprocessor-based platform.

systems, networking, and other emerging topics, including entrepreneurship, nanotechnology, and environmental issues.

Starting in the first introductory electrical and computer engineering (ECE) class, the students build the basic TekBot platform. With it, they learn basic circuit analysis techniques, digital logic operation, and bipolar transistor biasing. In the laboratory, as they assemble their TekBot, they apply basic circuit techniques that they were exposed to in lecture. In addition, they see real examples of analog circuits, digital circuits, and even analog-to-digital converters.

An exit survey administered to all students in the introductory class revealed the value of the TekBot and its associated laboratories. Students were asked to rate how important each of 13 different resources was to their understanding of electronics fundamentals. Of those resources, 64% of the students rated the TekBot laboratories as very important. This rating was only exceeded by laboratory teaching assistants (66%) and posted answers to homework problems (72%). The key here is how understanding is enhanced. Observations of laboratory sections indicated that students did use knowledge gained in lecture; however, the survey seems to indicate that the laboratory and teaching assistants were more important in developing a deeper understanding of that knowledge.

A series of interviews conducted with students in another class using the platform for learning curriculum revealed why the TekBot appears to be so important in developing understanding. When asked about the connections between lecture and laboratory, one student responded: "... right now what I am doing is I try to think back and I start remembering because I can remember physically what I was doing. And so that kind of helps with the concepts and when you are taking tests you are thinking, oh yeah, that is why I did that. I think that the TekBots really do help out with those concepts."

This course is followed by an introductory digital logic class where the students are introduced to combinatorial and sequential logic design in lecture. In the laboratory, they remove the analog controller board from the TekBot platform and construct a digital controller board that contains a complex programmable logic device (CPLD) [see Fig. 3(b)]. Each laboratory experiment involves designing logic networks based on the lecture material.

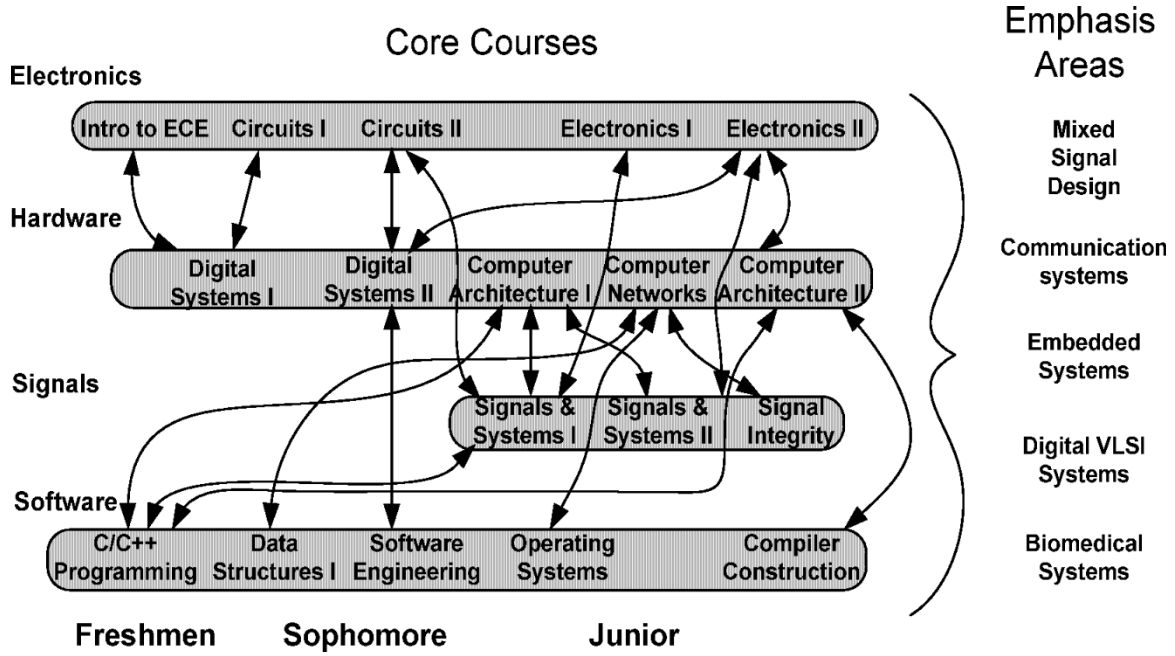

Fig. 4. Computer engineering curriculum illustrating the integration among topics.

Current software tools are used to design the logic. The configuration data is downloaded to the TekBot CPLD through a PC serial port. Once downloaded, the students can see first hand their designs in action. Students make an important connection between basic analog electronics and digital electronics toward the end of the term. In their final projects, they design a state machine that is downloaded to the CPLD that makes the robot exhibit the same behavior as the analog controller. This state machine flows naturally out of the lecture material where state machines and sequential logic are covered. Thus, the completed digitally controlled TekBot backs up, turns, and moves forward when it comes in contact with a wall.

Important to the laboratories are the challenge problems. They are optional problems that provide an innovation challenge to the advanced students, and they provide incentives for other students to push beyond their personal boundaries. An example of such a problem is an enhanced digital controller challenge. The original functionality is modified when the TekBot makes two contacts with the wall without traversing more than about a foot. The third time the robot hits a wall, it adjusts its turning radius so that it clears the obstacle.

Observations of a laboratory section in the digital logic class revealed that students were taking advantage of the challenges offered by the TekBots. It appeared that instances of innovation were often associated with increasing challenge. A pre- and post-survey conducted during this class supports these observations. Students were asked in both surveys how others might describe them in terms of a variety of traits related to innovation. At the end of the class, students enrolled in the laboratory believed they were more likely to be described as persons who could now come up with novel ideas than before they enrolled in the class. A number of students opted not to take the laboratory for the course, and they did not demonstrate this same shift [31].

Closely associated with the challenges provided by the TekBot is the development of a sense of community. The application of the content becomes increasingly sophisticated to the point where students find it important to interact with each other and with the laboratory teaching assistants in order to develop a solution to the challenge problem. This sense of community has not only been observed in laboratory sections, but also documented in student surveys. The study of students in the digital logic class showed an increase in mentoring and leadership between pre- and post-surveys [31]. The same increase did not exist for students not taking the laboratory. In the introductory class, 76% of the students felt they could exchange ideas with other students or the laboratory teaching assistants, which further indicates the development of a community.

In the sophomore year, students take Circuits I and II, Digital Systems II, and Data Structures and Software Engineering. Circuits I and II emphasize the steady-state and transient analysis of RLC circuits. These topics mesh well with Digital Design II where the emphasis is using hardware description languages (HDLs) to synthesize real logic gates while understanding their operation with realistic electronic parasitics effects, including interconnects, loading, and RLC effects of power supplies. The data structure and software engineering courses support these two courses by providing many of the software techniques and tools that make for reusable and correct HDL-based designs.

At the end of the sophomore year, the TekBot begins to transition from being a device with electronic components to a more sophisticated platform with "black boxes" on board. This transformation begins with the assembly language programming and introductory computer organization course. In this class, the CPLD board and the motor control board are replaced with a microcontroller board, as shown in Fig. 3(c).

The microcontroller drives the pulsewidth-modulated (PWM) inputs to the servomotors directly. The students learn

TABLE II
SUMMARY OF COURSE CONTENT FOR COURSES IN FIG. 4

Intro to ECE	System design concepts, Definitions of current and voltage, Circuit elements and schematics, Modeling with voltage and current sources, Kirchoff's laws, Analyzing circuits with SPICE, Analyzing transistor and diode circuits, Operation of logic gates, Simplifying boolean equations.
Circuits I	Sources and resistance, Kirchhoff's laws, Nodal and loop and mesh analysis, Source transformation, linearity and superposition, Thevenin's and Norton's theorem, Inductance and capacitance, Laplace transform in circuit analysis
Circuits II	Laplace transform review, Bode plots, Fourier sSeries, Fourier tTransform, Two-port circuits
Signals and Systems I	Differential and difference equations, convolution, Fourier transform, Frequency-domain analysis, Signal sampling and reconstruction
Signals and Systems II	Discrete-time Fourier analysis, The Laplace transform, Transfer function representation, The z-transform
Electronics I	Introductory semiconductor physics, Diode current-voltage characteristics, Bipolar junction transistor (BJT) characteristics, circuit design, Field-effect transistor (FET) characteristics, circuit design, MOS, and BJT differential amplifiers
Electronics II	Frequency response of BJT and MOS amplifiers, Amplifiers with feedback, Multistage amplifiers, Ideal operational amplifier circuits, Non-ideal op-amp effects, Op-amp filters and oscillators, Multistage op-amp circuits
Electronics III	MOS digital circuits, Waveform generation, Power amplifiers, Power supplies
Communication Networks	Layered architectures, Circuit and packet switching, ISO reference model, Point-to-point protocols, Eerror control, Framing, Virtual circuits, Datagrams, Routing, Congestion control, Reliable message transport, Internetworking.
Computer Architecture I	Basic computer organization, Assembly language programming, Instruction types and addressing modes, Subroutines. Assembler usage, Programming techniques.
Computer Architecture II	Performance Metrics, Instruction set architecture, CPU design and implementation, Pipelining, Memory hierarchy, cache and virtual memory, Input/Output design, interrupts, DMA.
Digital Systems I	Number systems, Boolean algebra and switching expressions, Canonical forms of logic functions, Circuit minimization via Karnaugh maps, Standard combinational modules, Arithmetic circuits and ALUs, Flip flops, Counters and shift registers, State Machine design, Programmable logic device usage
Digital Systems II	Logic design with hardware description languages (HDLs), Synthesis of HDLs, Area and timing optimization, Field programmable logic device usage
Advanced Digital Design	Design of digital cells for application specific integrated circuits (ASICs), cell layout, simulation and characterization, high speed digital design, clock distribution and skew
Digital VLSI Design	Design or large digital systems with an HDL. Specification of large systems. Top down design, testbenches, large system considerations.
Signal Integrity	Distributed-circuit effects in high-speed switching circuits, Methods of source and load termination, Crosstalk, electromagnetic interference, Switching and power distribution noise, Integrated circuit packaging
Embedded Systems	Hardware/software Interactions in embedded microprocessor systems. Interrupt latency, C/in-line assembly coding for speed, I/O peripheral protocols, software/hardware control of electromechanical devices
Computer Programming	Fundamentals of C/C++: Operators, Control structures, Functions, Data types, Pointers, Arrays, Strings, and structures.
Data Structures I	Space and time requirements, Abstract data types, Lists, Stacks, Queues, Trees, Queues and heaps, Hash tables, Searching and sorting
Software Engineering	Programming methodologies and tools, issues of style, testing, and maintenance.
Operating Systems	Operating systems history, File systems, Process state, Process control threads, Signals, Pipes, Virtual memory, C and Bourne shell
Compiler Construction	Attribute grammars, syntax-directed translation, lex, yacc, LR(1) parsers, symbol tables, semantic analysis, peep-hole optimization, Design of a simple compiler.

the basics of computer organization and assembly language programming in both the class and the laboratory. The microcontroller is programmed to operate the robot similar to the analog and digital robots with additional functionality. For example, the timers on the microcontroller can be used to determine the angle of attack to a wall. From this point, they can adjust the angle at which it turns away from the wall so that a parallel course to the wall is established. In lecture, these timers are discussed in detail giving students background and context to solve the problem.

Following this stage are two signals and systems classes. The same microcontroller board is used in these two classes as in the microcontroller class. However, sensors for detecting the robot's environment via infrared and sonar methods are

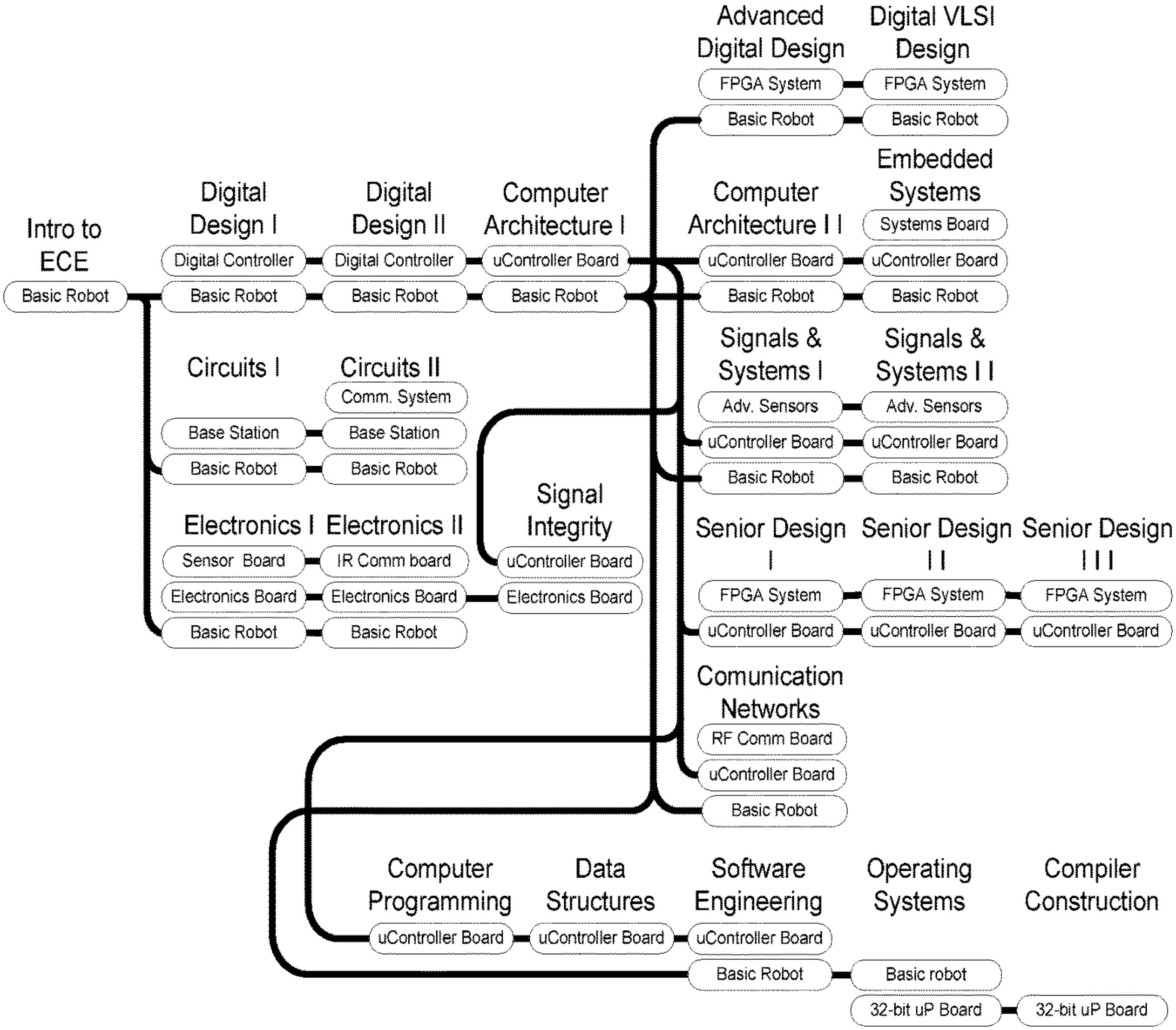

Fig. 5. TekBot platform evolution through the curriculum.

included. The infrared communication capability of the microcontroller board is used in this class to upload partially processed digital data to a PC from the sensors for further processing. Signal processing concepts taught in class are reinforced through the TekBot when the students collect real data and use the on-board microcontroller and Matlab toolbox to perform algorithms introduced in lecture.

The computer engineering classes that follow cover advanced logic design, computer architecture, embedded systems, VLSI design, and a senior design sequence. For these classes, another board, the advanced digital protoboard (ADP), is introduced. This board contains a 200 K-gate FPGA, flash and SRAM memories, and a configuration facility utilizing a universal serial bus (USB) connection to a PC. This board becomes a key subplatform that is used with the TekBot, as well as a stand-alone platform. For example, students may use this platform as an integral part of a senior design project.

In all fairness, sometimes the platform does not fit well with the class. For example, a device physics course may not map well to the platform. In this case, perhaps the best that can be done is to refer to the platform as an example of a product that uses the devices being described. The philosophy is that when the platform does not fit, it is not forced into the class.

Fig. 5 shows the transformation of the TekBot platform across the computer engineering curriculum. In each class, a particular hardware configuration is used to accentuate the lecture material, while keeping the basics of the robot base unchanged to maintain course continuity. Hardware changes are facilitated by the robot base design that allows different boards to be exchanged or added easily. In addition, this modular approach allows accommodation of students entering the program at virtually any point in the curriculum. They acquire the stripped down TekBot base and the particular boards used at that level. Fig. 3(a)–(c) shows an example of the evolution of the basic robot from the introductory ECE class, Fig. 3(a), to the digital design I, Fig. 3(b), to the computer architecture class, Fig. 3(c). After this class, the microcontroller board replaces both the analog control and the motor control boards. From this point, the FPGA board may be added with or without the microcontroller board intact.

V. CONCLUSION

One of the most pressing and critical needs for engineering graduates is to be natural innovators who are able to integrate their knowledge across many disciplines. While there have been many initiatives focused on reforming engineering education, none directly addresses the most pressing needs for educating future engineers.

In this paper, the authors present a holistic and systemic approach to engineering curriculum reform emphasizing innovation and integration of knowledge by employing a *platform for learning*. Introducing a platform for learning throughout the curriculum enhances the core engineering principles and provides a framework for integrating the engineering undergraduate experience. It serves as an umbrella under which various educational strategies can be used and links topics in the curriculum in a synergistic whole that reforms the curriculum.

REFERENCES

[1] W. A. Wulf, "The urgency of engineering education reform," *Bent Tau Beta Pi*, pp. 21–23, 1998.

[2] J. Bordogna. "Next generation engineering: Innovation through integration" (Keynote Address). presented at NSF Engineering Education Innovator's Conf., Apr. 8, 1997. [Online]. Available: www.nsf.gov/od/lpa/forum/bordogna/jb-eeic.htm

[3] (2002). Accreditation Board for Engineering and Technology (ABET). [Online]. Available: http://www.abet.org

[4] L. Carley, P. Khosla, and R. Unetich, "Teaching 'introduction to electrical and computer engineering' in context," *Proc. IEEE*, vol. 88, pp. 8–22, Jan. 2000.

[5] R. Manseur, "Development of an undergraduate robotics course," in *Proc. ASEE Frontiers in Education Conf.*, 1997, pp. 610–612.

[6] R. M. Felder, G. N. Felder, and E. J. Dietz, "A longitudinal study of engineering student performance and retention v. comparisons with traditionally-taught students," *J. Eng. Educ.*, vol. 98, no. 4, pp. 469–480, 1998.

[7] Does Collaborative Learning Work?. National Institute for Science Education. [Online]. Available: http://www.wcer.wisc.edu/nise/Research_Programs.asp

[8] D. W. Johnson, R. T. Johnson, and K. A. Smith, *Active Learning: Cooperation in the College Classroom*. Edina, MN: Interaction, 1998.

[9] D. W. Johnson, R. T. Johnson, and K. Smith, "Cooperative Learning: Increasing College Faculty Instructional Productivity," George Washington Univ., School Educ. Human Development, Washington, DC, ASHE-ERIC Higher Education Rep. No. 4, 1991.

[10] L. Springer, M. E. Stanne, and S. Donovan, "Review of Educational Research," University of Wisconsin-Madison, Nat.l Inst. Sci. Educ., Madison, WI, Research Monograph No. 11, 1998. Effects of cooperative learning on undergraduates in science, mathematics, engineering, and technology: A meta-analysis.

[11] B. Mashburn, B. Monk, R. E. Smith, T. Lee, and J. Bredeson. Experiences with a New Engineering Sophomore Year. [Online]. Available: www.foundationcoalition.org/publications/journalpapers/fie96/96 127.pdf

[12] D. Cordes, A. Parrish, B. Dixon, R. Borie, J. Jackson, and P. Gaughan. An integrated first-year curriculum for computer science and computer engineering. presented at Frontiers in Education Conf., Pittsburgh, PA. [Online]. Available: www.foundationcoalition.org/publications/journalpapers/ fie96/1048.pdf

[13] J. Richardson, J. K. Parker, and D. Cordes. The foundation coalition freshman year: Lessons learned. presented at Frontiers in Education Conf., Salt Lake City, UT. [Online]. Available: www.foundationcoalition.org/publications/ journalpapers/fie96/09 677.pdf

[14] M. Van Valkenburg, "Changing curricular structure," *Eng. Educ.*, vol. 79, no. 4, May/June 1989.

[15] E. Seymour and N. M. Hewitt, *Talking About Leaving: Why Undergraduates Leave the Sciences*. Boulder, CO: Westview , 1997.

[16] J. Wilson and W. Jennings, "Studio courses: How information technology is changing the way we teach, on campus and off," *Proc. IEEE*, vol. 88, pp. 72–80, Jan. 2000.

[17] R. J. Roedel, S. El-Ghazaly, and J. T. Aberle. An integrated upper division course in electronic materials and electromagnetic engineering-wave phenomena for electrical engineers. presented at Frontiers in Education Conf., Tempe, AZ. [Online]. Available: http://fie.engrng.pitt.edu/fie98/papers/1341.pdf

[18] R. Felder, "Reaching the second tier: Learning and teaching styles in college science education," *J. College Sci. Teaching*, vol. 23, no. 5, pp. 286–290, 1993.

[19] R. M. Felder and L. K. Silverman, "Learning and teaching styles in engineering education," *Eng. Educ.*, vol. 78, no. 7, pp. 674–681, Apr. 1988.

[20] T. Roppel, "An interdisciplinary laboratory sequence in electrical and computer engineering: Curriculum design and assessment results," *IEEE Trans. Educ.*, vol. 43, pp. 143–152, May 2000.

[21] Engineering Education Coalitions Websites [Online]. Available: http://www.eng.nsf.gov/eec/coalitions.htm

[22] B. J. Winkel and G. Rogers, "Integrated, first-year curriculum in science, engineering, and mathematics at Rose-Hulman Institute of Technology Nature, Evolution, and Evaluation," in *Proc. ASEE Annu. Conf.*, Champaign-Urbana, IL, 1993, pp. 186–191.

[23] J. Morgan, "A freshman engineering experience," presented at the 8th Annu. TBEEC Conf. Learning with Technology (TBEEC), Long Beach, CA, Nov. 1996.

[24] R. J. Roedel, D. Evans, M. Kawaki, B. Doak, M. Politano, S. Duerden, M. Green, and J. Kelly, "An integrated, project-based, introductory course in calculus, physics, english, and engineering," presented at the Frontiers in Education (FIE) Conf., Salt Lake City, UT, Nov. 5, 1996.

[25] K. Frair, "An integrated first-year curriculum: The foundation coalition and the University of Alabama," presented at the Frontiers in Education 25th Annu. Conf., Atlanta, GA, Nov. 1995.

[26] D. DiBiasio *et al.*, "Evaluation of a spiral curriculum for engineering," presented at the 29th ASEE/IEEE Frontiers in Education Conf., San Juan, Puerto Rico, Nov. 10–13, 1999.

[27] R. W. Lawler, *Computer Experience and Cognitive Development*. New York: Wiley, 1985.

[28] "Shaping the Future: New Expectations for Undergraduate Education in Science, Mathematics, Engineering, and Technology," Nat. Sci. Foundation, Washington, DC, 1996.

[29] "How People Learn: Brain, Mind, Experience, and School," Nat. Res. Council, Nat. Academy Press, Washington, DC, 2000.

[30] (1999) Employing a Productive Workforce. Nat. Sci. Foundation, Social Economic Sci. Div. [Online]. Available: http://www.nsf.gov/sbe/ses/sociol/works1a.htm

[31] D. Heer, R. Traylor, T. Thompson, and T. S. Fiez, "Enhancing the freshman and sophomore ECE student experience using a platform for learning," *IEEE Trans. Educ. (Special Issue on A Vision for ECE Education in 2013 and Beyond)*, vol. 46, pp. 434–443, Nov. 2003.

Roger L. Traylor received the B.S. degree in electrical engineering from Tennessee Technological University, Cookeville, in 1981 and the M.S. degree in electrical and computer engineering from Oregon State University, Corvallis, in 1991.

From 1981 to 1985, he was with GTE, Westborough, MA. From 1987 to 1996, he was employed with Intel Corporation, Hillsboro, OR. In 1996, he joined the School of Electrical Engineering and Computer Sciences (EECS), Oregon State University, as an Instructor. He has been involved with teaching very large scale integration (VLSI) design techniques, as well as introductory electrical engineering classes. He is also heavily involved with the development of the TekBots curriculum in the EECS. Having worked in the electrical engineering profession for more than a decade, he has a passion for welding textbook theory to practice.

Donald Heer received the B.S. and M.S. degrees in computer engineering from Oregon State University, Corvallis, in 2001 and 2003, respectively. He is also currently working toward the Ph.D. degree, conducting research in the area of integrated sensor systems using nanotechnology at the same university.

In 2003, he joined the School of Electrical Engineering and Computer Science, Oregon State University, as a Member of the Professional Faculty. In this role, he coordinates the TekBots program development and implementation. His research interests include creating innovative engineering education experiences.

Terri S. Fiez (A'82–M'85–SM'95) received the B.S. and M.S. degrees in electrical engineering from the University of Idaho, Moscow, in 1984 and 1985, respectively, and the Ph.D. degree in electrical and computer engineering from Oregon State University, Corvallis, in 1990.

From 1985 to 1987 and in 1988, she worked at Hewlett-Packard Corporation, Boise, ID, and Corvallis, OR, respectively. In 1990, she joined Washington State University, Pullman, as an Assistant Professor, where she became an Associate Professor in 1996. In fall 1999, she joined the Department of Electrical and Computer Engineering at Oregon State University as Professor and department head. She became Director of the School of Electrical Engineering and Computer Science, Oregon State University, in 2003. Her research interests are in the design of high-performance analog signal processing building blocks, simulation and modeling of substrate coupling effects in mixed-signal integrated circuits, and innovative engineering education approaches.

Dr. Fiez has been involved in a variety of IEEE activities, including serving on the committees for the IEEE International Solid-State Circuits Conference, the IEEE Custom Integrated Circuits Conference, and the International Symposium on Circuits and Systems (ISCAS). He was a Guest Editor of the IEEE JOURNAL OF SOLID-STATE CIRCUITS. She was previously awarded the NSF Young Investigator Award and the Solid-State Circuit Predoctoral Fellowship.