Web-based Personalized Laboratories for Engineering Students

Enid K. Sichel, *Member, IEEE* , Beverly Park Woolf, Mark Floryan

Abstract. We developed software that provides intelligent hands-on bench-top dynamic help to students as they study in laboratories for introductory circuit analysis. Tutoring help is available at "teachable moments" as opposed to students waiting days or weeks for traditional teacher-graded labs reports. Quantitative and qualitative studies show that using the software leads to improved learning, verbalization and conceptual knowledge and relieves teachers' workload. Students also reported that they enjoyed using the software.

***Index Terms*—circuit analysis, breadboard circuit, intelligent tutoring, adaptive intelligent support**

1. Motivation

Students do not retain much information from their engineering laboratories and are often unmotivated because they see labs as imposed hurdles, irrelevant to their planned career. Feedback is provided as written comments on student lab reports received often weeks later, far after optimal teachable moments, which occur during the laboratory (not after). Because of this stark disconnect between student action and focused feedback, students often make the same type of errors week after week. Additionally, engineering laboratories do not typically use efficacious forms of teaching, such as discovery-methods or project-based learning [1]. Understanding how people think and learn has forced a reconsideration of what is meant by effective teaching and has powerful implications for how instruction should be organized to elicit learning [2]. Today's students are less amenable to passive listening, less willing to read and more responsive to multimedia.

The

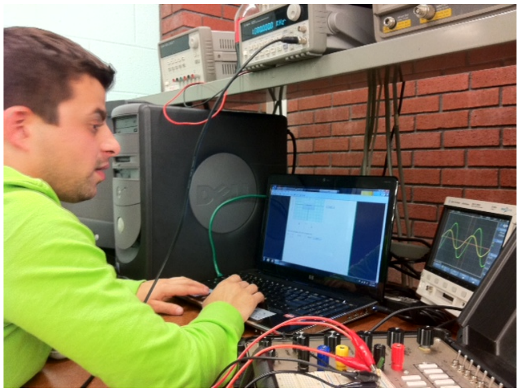

Figure 1. Students work in the software while engaged in physical lab activities. The student above sketches oscilloscope waves into the software (Figure 2) while recording phase shift and capacitive impedance on the oscilloscope (right). He draws VC, VR and V generator waves.

setting and student-to-teacher ratio in laboratories provide little time for faculty to focus on individual student needs and less time for analysis of student learning. Faculty are often absorbed checking student data and have little time to add new student experiences that might be important and relevant to industrial practice. This problem is shared by most science and technology curricula and delays integration of new topics and skills [3].

Simulations of labs are not an obvious solution, especially since they do not provide hands-on experiences. Guided use of sophisticated simulations before labs can enable exploratory and inquiry-based modes of learning [4]. Students are often oriented to "do the simulation" rather than reflect on and interpret what they see [5]. Our system attempts to address all of these issues simultaneously. The simulations do not always improve learning. For example, researchers at Purdue University evaluated the use of computer-simulation experiments in a senior-level chemical engineering course [6]. They found that the computer-simulated experiments led to better learning for some students, while others got more out of a traditional lab experiment. The authors caution against using instructional technology without evaluating its effectiveness.

Manuscript received February 14, 2014. This research was funded by a cooperative grant from the National Science Foundation, #0940967 and #1143659, "Authoring tool for a hands-on, on-line, lab curriculum for engineering technology students", Course, Curriculum, and Laboratory Improvement (CCLI) and REU #1324166. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

Enid K. Sichel is a consulting physicist. (e-mail: Enid_Sichel@verizon.net)

Beverly Park Woolf is a Research Professor in the School of Computer Science, University of Massachusetts, Amherst, 01002 USA (e-mail: bev@cs.umass.edu, 413-545-4265 or 413-545-1309).

Mark Floryan is a Lecturer in the Department of Computer Science at the University of Virginia. (e-mail: mrf8t@virginia.edu)

II. Instructional Software

The potential of computer-assisted instruction in engineering education is unarguable, but rigorous demonstrations of its effectiveness are in short supply [7]. Computers have begun to play a significant role in higher education beyond functioning as high-tech typewriters and calculators and have been widely used to enhance traditional curricula [7]. However, the literature shows that they have been scarcely used in the engineering curriculum. There is an opportunity for computer-based teaching tools in engineering and industrial technology programs [8]. In the chemical engineering curriculum, for example, courses have incorporated increasingly complex and realistic examples through the use of spreadsheets, mathematical and process simulation software, multimedia courseware, and resources available through the World Wide Web.

Well-designed instructional technology can facilitate learning in ways that cannot be achieved in traditional classroom settings [7]. Strong instructional software can personalize visual and verbal material but weak software continues to present overwhelming textual content of lectures [7]. The goal is software that customizes mentoring for individual students, supports them to take an active role in the learning experience by providing questions, problems, and invites them to explore what-if scenarios through simulations. Perhaps most importantly, software can give the students immediate positive or corrective feedback to their responses in a completely non-threatening and non-embarrassing manner.

One example in chemical engineering is where students used instructional software that included interactive instructional tutorials, a user-friendly equation-solving program, a physical-property database and a visual encyclopedia of chemical engineering equipment [7]. Results indicate that students did not use the software much. Even though the tools had great potential none of the 102 students reported using the courseware "frequently." A weak correlation was found between the frequency of use of the courseware and its perceived helpfulness in course performance (Pearson coefficient R=0.33). The implication is that providing unfamiliar instructional software in a time-consuming engineering course and counting on the students to make use of it on their own is not an effective strategy.

In another study, advanced manufacturing students tested the effectiveness of courseware about modular fixturing [8]. Previous research determined that students considered the courseware useful and were happy with the format and the user interface [9]. During the semester, the traditional lecture on the subject of modular fixturing was replaced with the software tutorial. After all of the students completed the tutorial (at their own pace) the students took a post-test. The complex components used in tooling applications and the subtle differences in their application make computer graphics based instruction an attractive teaching tool. This limited case study involved advanced manufacturing students in industrial technology and has shown that students achieved an increase

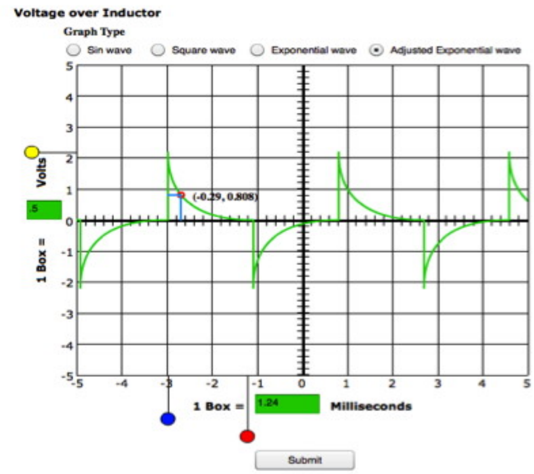

Figure 2. Students use the CIRCE software to sketch electronic waves. Three software levers (bottom and left) are used to fine-tune the wave's period (red), offset (blue) and amplitude (yellow). This RL transient lab moves students into the part of the course where simple algebra no longer suffices for understanding the concepts.

in their knowledge of modular fixturing concepts. Further statistical tests show that the dependent sample t-test shows significance between pre and post-test while the independent sample t-test failed to show significance. The data are a bit ambiguous, but the independent t-test (observed) statistic is not drastically lower than that of the t-test (critical value), warranting further research.

Issue to Address. Supporting student learning in engineering laboratories through software raises several issues, including institutional and technical barriers, e.g., policies that may hinder work in classrooms or structural

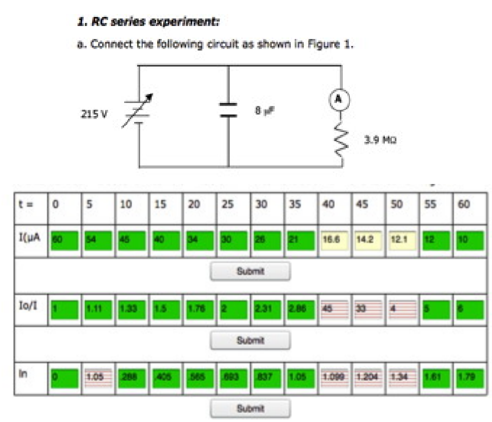

Figure 3. Students record the current at 5 second intervals and the software displays green highlights for answers within the allowed range, stripped red for data outside that range and yellow after three tries, at which point the software enters a calculated answer.

issues (Internet connection), and content mismatch for some students. Most two and four year colleges are constrained by their laboratory equipment. Typically, each organization has a collection of equipment that must be used unless a windfall is received to re-equip the laboratory. Commercial laboratory manuals are of limited use in a specific laboratory, as the organization must be equipped with the apparatuses designated by the laboratory manual. This research is also constrained by the physical realities of the lab, e.g., voltage not high enough, oscilloscope results unclear. Oddly enough, we have not had any serious equipment problems at the schools where this software was tested, nor problems with sufficient lab bandwidth. Problems with d.c. microammeters or high-quality inductors (~$50 each) might lead to new purchases.

Other issues occur at the level of the type of laboratory equipment used, for example, a lab might require a sensitive microammeter that is not available. One *solution* is to increase the value of the physical (and software) voltage. Another issue can be identified as the *'data are data'* problem and refers to the fact that the software accepts any input for *measured* (not calculated) values, e.g., any data taken from a meter are accepted as valid. This can lead to later problems, especially if "measured" values were off by powers of ten. One *solution* is to check the magnitude of measured input and advise students about whether a value is off by powers of 10. However, this solution may break down when students are correctly reading values from faulty equipment and becoming confused by the software's lack of flexibility.

Another issue is how many attempts students should have to input data into software. One approach is *three strikes and you're out.* Students might be allowed three input answers, after which the system submits a calculated answer. However, after the system fills in answers, students can't change their input values, which can be frustrating. Clearly *dependencies exist among answers*, e.g., subsequent questions depend on prior responses and incorrect answers on one page will escalate to cause large errors later on. Should the software adjust the expected student responses to later questions based on the actual submitted values on dependent questions? The need for dependencies causes other changes to the student's conceptual model that make the software seem fickle. For example, a student may try to readjust problem A in the system. However, if problems B, C, and D all depend on A's value, then system might force a student to redo his or her work on problems B, C, and D. This can often surprise the unsuspecting student, leading to frustration.

Another issue is the human-computer interface. We know that the software must be friendly and accommodating to all students. When it is not and students complain, developers need to react quickly to modify the software. In the system described in this paper, students used Tweets to communicate to researchers about the quality of the software. This was unexpected, but welcome in that it pointed out flaws in the software. Other students used Tweets to remark on the usefulness of the software in completing the lab. These changes to the software through the formative assessment process in Fall, 2012 appear to have been successful. In the three labs that occurred after these student statements, no student Tweets remarked (negatively or positively) on the functioning of the software itself. In this way, Tweets became a useful tool – helpful, but not calling attention to itself.

An important barrier to the widespread use of most traditional computer-aided engineering teaching tools is that they require increased faculty time including specialized faculty training. The software described in this paper, requires very little faculty training and provides authoring tools for faculty to customize labs to their own content (not discussed here). We discovered that one impediment to use of this software is faculty who have invested a great deal of effort in writing their own laboratories and are not willing to take on new software.

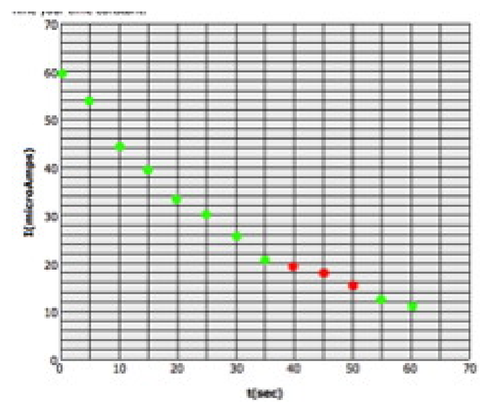

Figure 4*. Current in a discharging RC circuit. Students transfer data to graphs and draw conclusions, e.g., the time constant. They plot data points and receive immediate feedback about acceptable (green) or out of acceptable range points (red). Skills include algebra, knowledge of units, and familiarity with new words and symbols, e.g., micro and μ.*

Intelligent instructional software for engineering education. Intelligent instructional software models the student's knowledge and behavior and then adapts the software to the learning needs of individual students. Thus it might produce advanced material for a high achieving student and remedial material for a struggling student. One of the best examples is CyclePad, a fully implemented articulate virtual laboratory that uses qualitative reasoning to capture thermodynamic knowledge from an introductory textbook [10]. It provides explanations of calculations and coaching support for students through a major portion of a semester's training in thermodynamics, including key properties of systems that interconvert work and heat, such as power plants, propulsion systems, refrigerators, and heat pumps. This system provides numerous simulations and interaction but students do not work with a physical system and do not receive immediate feedback for their hands-on activities.

Other intelligent tutors generate natural language dialogues with students and might use deep parsers and generators, together with systems that reason about domain knowledge and diagnose student problems, to produce detailed analyses

Figure 5. Students collaborate to answer the software questions related to circuit analysis.

of student utterances and generate automatic feedback. CycleTalk, built upon the CyclePad simulation environment (above), investigated novel ways of using tutorial dialogue to teach thermodynamics [11]. The system empirically evaluated dialogue that invited students to negotiate problem-solving goals. For example, the system asked student to "do you think it is a good idea for the . . . temperature to be increased and kept high?" or "What happens to the steam quality if you increase the maximum temperature?" Empirical results provide strong evidence in favor of tutorial dialogue. BEETLE II is another tutorial dialogue system, this time to teach circuit analysis, whose interface included an area to display reading material, a circuit simulator, and a dialogue history window [12]. All interactions with the system were typed. Students read curriculum slides and carried out exercises that involved experiments with a circuit simulator and explained the observed behavior. The system also asked high-level questions, designed to accept unrestricted language input.

Another intelligent software platform helps automate collaborative learning experiences while teaching mechanical engineering to undergraduates [11]. These tutorials teach fundamental concepts through dynamic dialogues. In a typical assignment, students perform a design or modeling task with commercial software. As students collaborate electronically, an intelligent agent monitors their interactions and interjects questions or comments in response to the use of key phrases, or due to other triggers. Again this software does not provide hands-on experience, though it does monitor immediate dialogues between members of student teams.

III. A CYBERLEARNING LABORATORY

We developed a suite of Cyberlearning laboratories; instructional software that uses *intelligent models* to teach engineering labs, support hands-on learning and explore the impact of computational labs on student learning and affect. CIRCE, circuit analysis computer environments, keeps

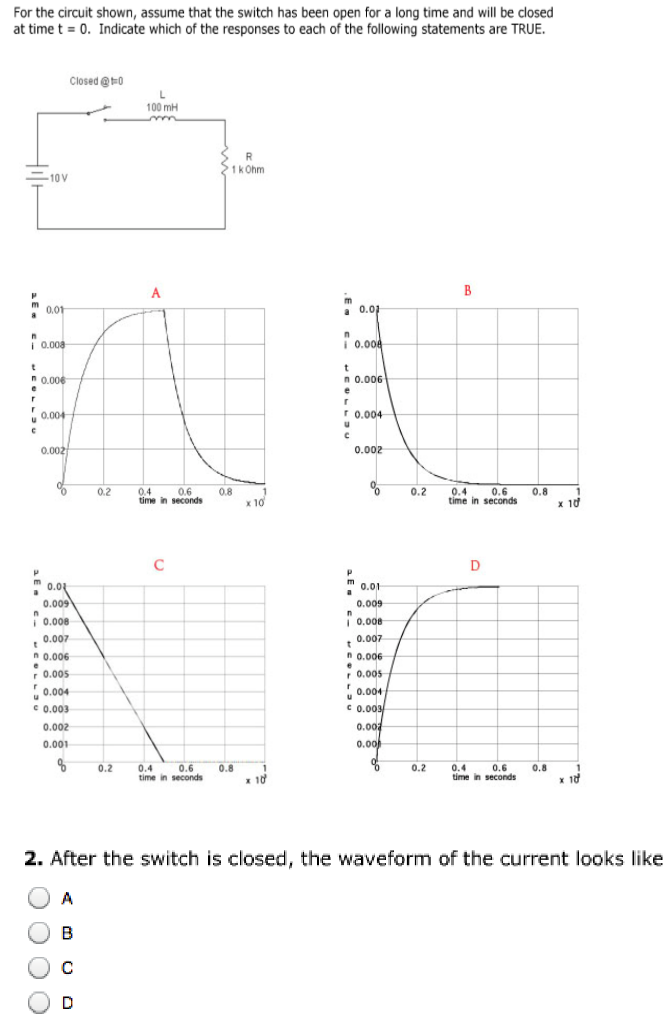

Figure 6. Conceptual questions for transient voltages in an RL Circuit. This conceptual problem is from the conceptual inventory of Rancour and Helgelard, UMass-Dartmouth, see www.foundationcoalition.org.

students on task during typical laboratories with *regular feedback* and provides *engineering content, see Figures 2,3,4, 6,and 7*. It provides *immediate student feedback, real-time error correction* and *final student grades.*

It comes as no surprise that students who receive immediate feedback perform better on exams. CIRCE *tracks student data in real-time* and identifies activities that actually lead to learning gains. Students perform their usual hands-on experiments or projects using physical circuitry while submitting data to the computer in real time. For example, they receive immediate feedback as they sketch oscilloscope waves into the software. As they plot a graph or record data or tables they receive feedback about whether the data is within the expected range (green) or outside the expected range (red), see Figures 3-4. Out-of-range data points are immediately flagged.

We developed and evaluated seven introductory electronic engineering laboratories for freshmen and sophomore electrical engineering students. These labs are *completely vetted* and *freely available.*[1]

IV. Adaptive Personalized Hints

The tutor provides individual adaptive responses that provide appropriate help and hints, based on an evolving model of the student's presumed knowledge and emotion. Individual students can move ahead or return to prior topics as they want. The pedagogical companion provides content-based responses to students, e.g., "Your answer is off by powers of 10." or "Please check your amplitude in the oscilloscope graph," (similar to Figure 7).

The presence of an instructional coach who cares, or, in the case of CIRCE, who appears to care, can provide a more personal experience and help students to persist at a task. The intelligent tutor helps students diagnose probable causes of mistakes.

The coach first initializes a decision tree model of proper reactions to various situations that can arise within the tutor. This model is built using data from a shared spreadsheet that is maintained by subject matter experts. The spreadsheet contains both specific information regarding proper reactions to student input, as well as data that generalize across the entire system.

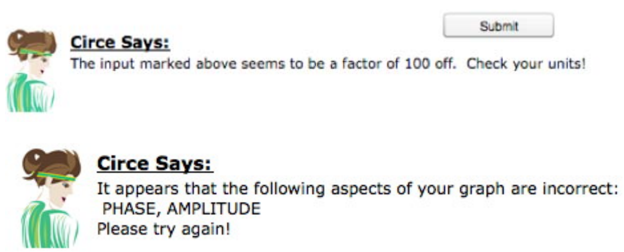

Figure 7. Learning agent and personalized help. The pedagogical agent in CIRCE provides context-based hints to help improve students' lab performance.

The coach's model can be updated from this spreadsheet at any point in time, representing a step towards a truly evolving coach, one in which knowledge updated in the data source is consistently uploaded and incorporated into the tutor automatically (though periodically). Actions that students make within CIRCE are fed through this decision tree and if necessary, feedback displayed to the student. The coach is not limited by the amount of feedback given, and will often provided several suggestions that may span different activities within the tutor. The feedback is presented in a way that makes clear to students 1) the total amount of feedback available at any time, and 2) the specific elements of the tutor (chart, table) with which each feedback is associated (for clarity).

[1] CIRCE is available at http://althea.cs.umass.edu/circe/enid/guestlogin_new.php. Give yourself a guest account and test out the labs.

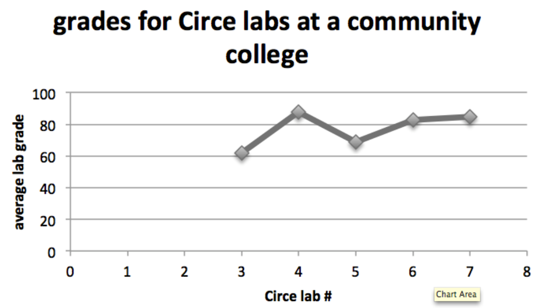

Figure 8. Lab Grades. Median freshman lab scores at the community college. Freshman lab scores were relatively low (~ 60%) at the beginning of the semester and grades improved as the semester progressed. All scores were calculated within the CIRCE software. Data were not available for Labs 1 & 2.

V. Evaluation Results

We performed both quantitative and qualitative analysis of student learning using the instructional software.

We explored student comprehension of key engineering concepts and whether students demonstrated increased positive affect conducive to learning (e.g., increased interest, engagement) while using the system. The data analysis in this paper was acquired from students who received feedback from an earlier version of the coach that contained nearly ten times *less* information than the current version (at the time of this paper's writing). We cannot present results on the efficacy of the coach explicitly, as there were no control groups for comparison. In the near future, we will present results regarding the efficacy of the coach trained from a rich and plentiful data source. However, even the more basic coach implementation provides relevant, on-time feedback to students. Comprehension of key engineering concepts was positively improved and students demonstrated increased positive affect conducive to learning (e.g., increased interest, engagement).

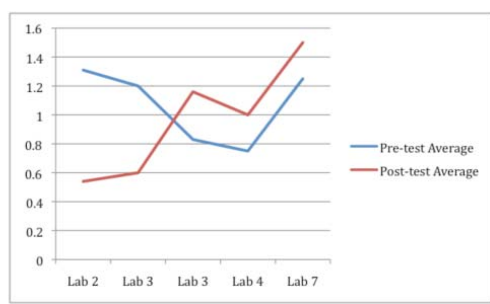

Figure 9. Student conceptual knowledge as measured by pre and post tests. Conceptual knowledge was measured by conceptual problems, see Figure 6. Post knowledge improved as the course and labs progressed. Pre-tests shown in blue and post-tests in red.

We evaluated student *comprehension* of key concepts, ability to *verbalize concepts* and *affect* (e.g., interest, boredom, engagement). One goal was to evaluate whether providing rapid responses during labs is differentially associated with gains in reasoning skills. Research suggests that learners benefit when their thinking is made visible and when they take responsibility for their own answers rather than wait for faculty approval. The labs were evaluated at both a large Southwestern university and a large Northeastern urban community college. Unfortunately, data collection at the university was problematic, in that a new lab was written from scratch and the team was unable to assist during the lab to handle the inevitable bugs. Results from the community college are discussed below. We used four evaluation tools (both qualitative and quantitative) to assess student response, including a pre-lab and post-lab physics quiz in concept inventory style, laboratory grades, student Tweet (format 280 characters) and a survey to assess student attitudes.

Improved ability as measured by lab grades. We evaluated 78 lab grades assigned by the software and based on recording student answers and key strokes in Fall 2012 at the Northeastern community college. These data include results for all activities, e.g., student input to graphs, tables and sketches of waves from oscilloscopes, see Figures 2-4. By recording all student activity within the software, analysis of student data is efficient and easy to collect. Teachers do not need to evaluate lab reports. The grades of freshmen increased as the semester and labs progressed; sophomores received generally higher grades, in the 80% range. Of course, because the software assigns the grades, we cannot necessarily claim that this is evidence of learning. Also other confounds could be responsible for grades including difficulty of individual labs which could explain a sharp decrease for lab 5. We do, however, posit that this is reflective of students becoming more acclimated to the system.

Improved conceptual knowledge as measured by pre-post quizzes. We evaluated 41 pre and post conceptual quizzes associated with each lab unit. Traditional tests are not necessarily effective tools for diagnosing the content and structure of students' knowledge of physics or engineering [12]. Instead conceptual quizzes are often used to measure a student's conceptual and qualitative knowledge, which refers to relationships and flow among objects and is needed for proficient problem solving. Figure 6 is an example of a conceptual quiz for transient voltages in an RC circuit. Additionally, particular types of cognitive processes are required for acquiring conceptual knowledge and building useful knowledge structures. We used quizzes based on a concept inventory of circuit analysis problems developed by David Rancour and Bob Helgelard at UMass-Dartmouth[2]. We selected two questions from this inventory for each circuit pre and post-quiz. Grades for pre and post quizzes improved as the labs and semester progressed, see Figure 9. Most quizzes required a circuit diagram or computer graphic. They did not count in the student's grade and provide only right/wrong feedback and one try only.

Several pieces of evidence indicate that CIRCE has a learning curve, during which student learning may actually suffer. We saw that grades rose over time as students progressed through the CIRCE labs, which may imply learning, but most likely seems to show an increased level of comfort and experience with the cyberlearning environment. Our pre and post test analysis shows results that are consistent with this hypothesis. Later labs resulted in apparent learning from pre to post test. However, the earlier labs actually show the opposite effect. We believe there are two primary reasons for this. Firstly, as stated above, comfort with the system and cyberlearning lab paradigm likely had an effect on student learning, which corrected itself in later labs. Additionally, the difficulty of each lab increases over time, resulting in more opportunity for the system to have an effect on student learning. We believe that our quantitative results are explained by some amalgamation of these two effects.

[2] The concept inventory of circuit analysis problems was developed by David Rancour and Bob Helgelard, from UMass-Dartmouth, and attributed to them in each CIRCE Lab, see www.foundationcoalition.org.

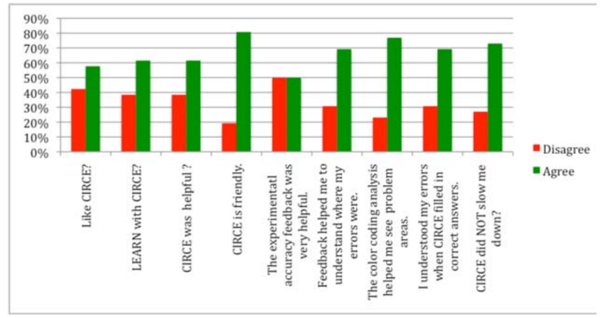

Figure 10. Positive affect after using software. Survey results for interest and engagement for all large urban community college data sets. Mean student responses during this research. Students enjoyed using the software and thought it was friendly, helpful, and enabled them to understand their errors. Positive responses (4-6) in green and negative responses (1-3) in red.

Before/ After Knowledge	Example Student Tweet
1.Ambiguous/ Incomplete	*(pre)* "in a series circuit"; *(post)* "My Mind is blown"
2. Confusion/ Confusion	*(pre)* "Current directly proporonal to resistince"; *(post)* "current changes with different amps"
3.Confusion/ Understanding	*(pre)* " We know nothing;" *(post)* "We learned how to use a function generator"
4.Understandi ng/Understand ing	*(pre)* "it discharges quickly to start off then gets slower and slower as time goes on unill it reaches zero *" (post)* " I learned how to calculate Io/I and ln (Io/I) and how to plot them."

Table 1. Specific examples of student 'tweets' and their respective classifications under our analysis.

Positive student affect as measured by surveys. One of our goals is to develop software that drives innovation in the way instructors teach or students learn. The students enjoyed using the software; they thought it was friendly, helpful, and enabled them to understand their errors. This evidence comes from surveys, focus groups, and observed changes in teaching practice. Results from surveys from Fall 2012 at a large urban community college are shown in Figure 10. Twenty-six students completed the Likert scale survey (6 strongly agree; 1 strongly disagree). Additionally, the software seemed to help students pay more attention to details.

Improved verbalization as measured by Tweets. We evaluated 70 student Tweets, brief textual messages crafted by students within the system to describe their knowledge about engineering topics, see example Tweets in Table 1. The goal was to identify students' qualitative knowledge about general topics (e.g., inductance, capacitance) before and after each lab.

For example, a pre-lab Tweet asks "What do you know about discharging a capacitor through a resistor. What happens in a one time constant?" and the post-lab Tweet asks "What did you learn in this laboratory?" Tweets were classified into four categories: ambiguous; confusion before & after; confusion before/reasonable understanding after; and good understanding before/reasonable understanding after. Table 2 presents the number of raw Tweets analyzed and Table 3 provides example summary comments from students.

The Tweets are a narrative summary of what students understand before the lab and an assessment by the student of what he/she learned in the lab. Even with an understanding of Tweet shorthand, the students' responses were hard to sort because of incomplete responses and undeciferable responses

	Tweets from Community College Students					SUM
Tweet "bins"	**Lab 3**	**Lab 4**	**Lab 5**	**Lab 6**	**Lab 7**	**Sum**
1. Ambiguous or incomplete	2+11=13	1	10	1	2	27
2. Confusion or no knowledge before lab and confusion after lab	1+1=2	0	1	0	0	3
3. Confusion or no knowledge before lab & reasonable understanding or statement of satisfaction after lab.	1+1=2	1	0	0	0	3
4. Good understanding pre-lab and reasonable understanding or statement of satisfaction after lab	6+4=10	2	7	3	2	24
Sum	27	4	18	4	4	57

Table 2. Raw Tweet data. The high numbers in Bin 1 row 1 (Ambiguous) include Tweets that were not decipherable and times that students failed to fill in any response, either pre-lab or post-lab.

.

VI. DISCUSSION AND FUTURE WORK

Evidence presented in this paper provides a powerful argument for building Cyberlearning tool for science laboratories. One goal is to develop software that drives innovation in the way instructors teach and students learn. These environments provide an opportunity to increase student learning and affect, while simultaneously freeing instructors from the shackles of tedious administrative tasks such as grading lab reports and checking lab data. Of course, teachers still need to observe student achievement and intervene perhaps with more individual mentoring. CIRCE provides tools (not discussed here) for instructors to view the work and assessment for all students. These tools allow further freedom from administrative tasks while also providing clear assessments of student performance.

Additionally, an opportunity exists for intelligent tutoring systems to optimize not only learning, but also long-term attitudes related to students' emotions while working on science labs. By modifying the "context" of the tutoring system including students' perceived emotion around engineering, intelligent tutors might optimize and improve their engineering attitude. Prior results showed that a student's value of a topic and self-concept in that topic can predict long-term success, e.g., students who value engineering and have a positive self-concept of their engineering ability might perform better [13]. This paper argues for the use of cyberlearning environments, in which intelligent software is used hand-in-hand with physical laboratories to increase student learning and relieve teachers of administrative duties. CIRCE invites students to submit responses to seven circuit analysis laboratories. Studies show that students initially seemed to receive little value from the CIRCE environment,

Table 3. Example knowledge and emotion expressed by students in CIRCE pre and post lab Tweets. During the research, students used their creative spelling to provide unsolicited comments about their experience

Student Remarks about their Experience
"I learned how to do the calculations and i like using the computer to put in the answers and finding out if they are right or not makes it easer to do and understand"
"The diagrams helped me better how too hoom up my ammeter and voltmeter too the circuit"
"i learned that the test is tricky about what numbers you put in."
"I learned how to do the calculations and i like using the computer to put in the answers and finding out if they are right or not makes it easer to do and understand" "my measurements were off at first but then i used circe to check my answers"

potential evidence of a necessary learning curve for the software and possibly some mistrust (some valid, some not, as evidenced by the student Tweets). However, later labs provide evidence of increased learning both quantitatively and qualitatively. Overall, student's affect was positive after using the software. In the near future, we look forward to using the new version of the coach, which has been trained from a rich and plentiful data source and contains ten times the amount of response messages. We will also present results regarding the efficacy of the coach. We intend to continue exploring the use of cyberlearning environments in science and engineering laboratories.

ACKNOWLEDGMENT

We acknowledge Professor Florence Sullivan of The College of Education, UMass, who performed the evaluation studies, advised us on evaluation tools and we are grateful for her advice on Tweet-style narrative evaluation, pre- and post-lab physics concept inventory quizzes, and affect-measurement questionnaires. UMass student William Ryan began coding the website as a student project and we thank him for his early participation. EKS thanks Five Colleges, Inc. of Amherst, MA for providing her an academic home during this grant. Our thanks go to Professors Jeffrey Cooper and Eliano Soares of Springfield Technical Community College and Professors Elizabeth J. Brauer and David Robin Scott of Northern Arizona University for their participation

REFERENCES

1. Brown, A. L., & Cocking, R. R. (2000). *How people learn*. J. D. Bransford (Ed.). Washington, DC: National Academy Press
2. How People Learn: Brain, Mind, Experience and School (2000); NRC Commission on Behavioral and Social Sciences and Education; National Academy Press
3. National Science Foundation (2006). *Report of NSF Midwest Region Workshop on Integrative Computing Education and Research (ICER) Preparing IT Graduates for 2010 and Beyond*, NSF CNS, February.
4. National Science Foundation (1998) *Information Technology: Its Impact on Undergraduate Education in Science, Mathematics, Engineering and Technology*, Report on an NSF Workshop, NSF DEHR/DUE, NSF Report 98-82.
5. National Science Foundation (1999). *Workshop on Improving Undergraduate Education in the Mathematical and Physical Sciences through the Use of Technology*, Report on an NSF Workshop, NSFMPS/DUE/HER, July 20-22.
6. White, S. R., and G. M. Bodner. (1999) "Evaluation Of Computer Simulation Experiments In a Senior Level Capstone Chemical Engineering Course." *Chemical Engineering Education* 33.1.
7. Michel, A., Felder, R. M., Genzer, J., & Fuller, H. (2000). Student Use of Instructional Technology in the Introductory Chemical Engineering Course. In *2000 ASEE Annual Conference Proceedings*.
8. Haig Vahradian, and O'Charoen, 2006, A Modified Case Study: Teaching Modular Fixturing Concepts by Integrating Multimedia Courseware, *Proceedings of The 2006 IJME - INTERTECH Conference.*
9. O'Charoen, V., Hall T.J.K, and Vahradian, H. (2005). Modified Case Study: Using Multimedia Courseware To Teach Modular Fixturing. ASEE Annual Conference & Exposition
10. Proceedings 2005 Forbus, K. D., Whalley, P. B., Everett, J. O., Ureel, L., Brokowski, M., Baher, J., & Kuehne, S. E. (1999). CyclePad: An articulate virtual laboratory for engineering thermodynamics. *Artificial Intelligence*, *114*(1), 297-347.
11. Dzikovska, M. O., Moore, J. D., Steinhauser, N., Campbell, G., Farrow, E., & Callaway, C. B. (2010, July). Beetle II: a system for tutoring and computational linguistics experimentation. In *Proceedings of the ACL 2010 System Demonstrations* (pp. 13-18). Association for Computational Linguistics.
12. Kumar, R., Rosé, C., Aleven, V., Iglesias, A., & Robinson, A. (2006, January). Evaluating the effectiveness of tutorial dialogue instruction in an exploratory learning context. In *Intelligent Tutoring Systems* (pp. 666-674). Springer Berlin Heidelberg
13. Beuth, J. L., Rosé, C. P., & Kumar, R. (2010). Software Agent-Monitored Tutorials Enabling Collaborative Learning in Computer-Aided Design and Analysis. In *Proceedings of the ASME 2010 International Mechanical Engineering Congress & Exposition (IMECE2010), Vancouver, British Columbia, Canada*.
14. Gerace, W., Dufresne, R. J., Leonard, W. J., & Mestre, J. P. (2001, July). Problem solving and conceptual understanding. In *Proceedings of the 2001 Physics Education Research Conference* (p. 33). Physics Education Research Conference, Annual Conference.
15. Royer, J. M., & Walles, R. (2007). Influences of gender, motivation and socioeconomic status on mathematics performance. *Why is math so hard for some children. Baltimore, MD: Paul H. Brookes Publishing Co*, 349-368.