THE DEVELOPMENT AND DEPLOYMENT OF A VIRTUAL UNIT OPS LABORATORY

SREERAM VAIDYANATH, JASON WILLIAMS, MARCUS HILLIARD, AND THEODORE WIESNER
Texas Tech University • Lubbock, TX 79409-3121

The success of engineering education relies heavily on training a student to apply the theoretical knowledge gained to practical situations. In traditional pedagogy, the theoretical element is typically provided through classroom lectures and tutorials. For the practical part, engineering laboratories play the major role. Luis Ando, *et al.*,[1] call this "learning by doing."

According to Hansen,[2] only 25% of what students hear stays with them, about 45% of what students hear and see is retained, and about 70% of what they do is retained when they use "learning by doing." Douglas Cooper[3] mentions that "such practice is motivating, promotes critical thinking, facilitates understanding in the use and limitations of the theory, and helps prepare students for challenges of the professional world."

Even though theoretical and practical knowledge are equally important, they come at different costs. In conventional training, the theoretical knowledge is imparted through classroom lectures and tutorials. This knowledge is relatively affordable. The practical component, which requires a laboratory setup, comes more expensively. The costs incurred include procurement of equipment, setup, maintenance, operation, and training. Moreover, the laboratory equipment is typically available only for limited time periods.[4]

BENEFITS OF COMPUTER-BASED EXPERIMENTS

Kadiyala and Crynes[5] published an exhaustive overview of computer-based instruction in engineering over the past 15 years. Reviewing 760 reports, they found convincing evidence that information technologies can enhance learning when the pedagogy is sound, and when there is a good match of technology, techniques, and objectives. The use of computer-based laboratories can also contribute significantly to reducing the costs of practical training.[6]

Students can access the computer-based laboratories more easily than accessing physical labs. In fact, every computer that can run lab software becomes a lab. With the proliferation of laptop computers, it is literally "lab anywhere." Also, a student can "redo" the experiments at home and try out new ideas as soon as he or she conceives them. In a traditional physical environment, this would be difficult, as the student must wait for the laboratory classes. Additional time must be spent to reconfigure equipment to explore alternate scenarios. Moreover, in many cases, the enthusiasm to try out new things diminishes with time. Therefore, the computer-based laboratory caters to the realization of "flashes of thought" that could happen anytime.

***Sreeram Vaidyanath** is a software engineer with Microsoft in Bellevue, WA. He completed a Master's of Science degree in Computer Science at Texas Tech University in December 2005, specializing in image registration. Prior to coming to Texas Tech, he earned a Bachelor of Technology degree from the University of Calicut, Kerala, India. His home is Palakkad, India.*

***Jason L. Williams** is a doctoral student in chemical engineering at Texas Tech University. Prior to entering the TTU doctoral program, he earned a B.S. in chemical engineering from TTU in May 2002, during which time he also contributed to the Virtual Unit Operations Laboratory.*

***Marcus Hilliard** is a doctoral student in chemical engineering at the University of Texas in Austin. Prior to entering the UT doctoral program, he earned a B.S. degree in chemical engineering from TTU in May 2002, during which time he contributed to the Virtual Unit Operations Laboratory.*

***Theodore F. Wiesner** is an associate professor of chemical engineering at Texas Tech University. His research interests are computer-based instruction, biomedical engineering, and bioprocess engineering. Prior to entering academia, he worked in the chemical process industries in the areas of polymer manufacture and wastewater treatment. He earned his B.S. degree from Kansas State University, his M.S. from the University of Houston, and his doctorate from the Georgia Institute of Technology, all in chemical engineering. He is a member of American Institute of Chemical Engineers, the American Society of Engineering Education, and the Biomedical Engineering Society.*

Little wonder, then, that computers are increasingly being deployed in industry.[8-11] Flowsheeting software, such as ASPEN or CHEMCAD, is used to design new processes and simulate existing facilities. Importantly, and perhaps more frequently, computer technology is used to collect and analyze data from operating processes to optimize them. Immediately after graduation, an engineer is most likely to work in an environment where he or she monitors and perhaps controls the physical equipment from a computer screen. Yet Sorby, *et al.*, in Reference 7, state that, "The engineering curriculum has evolved over the years to include some computer applications, but computer techniques for the most part are not an integral and pervasive part of the curriculum as they are in the industrial sector. To develop students who will succeed in their engineering careers, it is important that they be introduced to computer techniques early in their educational programs and that these skills are continuously used and reinforced throughout the curriculum." Hence, training students to collect data via computer interfaces prepares them to serve in such anticipated industrial environments.

Further, computer-based experiments provide a safe venue for students to experiment in understanding what might happen if they do not conform to the standard operating procedure and normal operating conditions. Some experiments, such as the runaway of an exothermic reaction, are difficult to treat cost-effectively and safely in a physical laboratory. In a computer-based lab, however, a student can safely explore such scenarios and be better prepared for real-world situations. Computer-based experiments can likewise actually train students to run the physical equipment in the same or subsequent courses. This is the same rationale behind training pilots in ground-based flight simulators prior to having them fly in real aircraft. In fact, several chemical processing firms have implemented training on dynamic process simulators for their plant operators prior to the actual running of a facility. Such virtual training can be of value to new engineers as well.

In this paper we report on our experience in developing and deploying a computer-based laboratory[1]. We conclude that, while the up-front investment in terms of labor in virtual laboratory development is substantial, the benefits of augmenting a physical laboratory with computer simulations can justify the development.

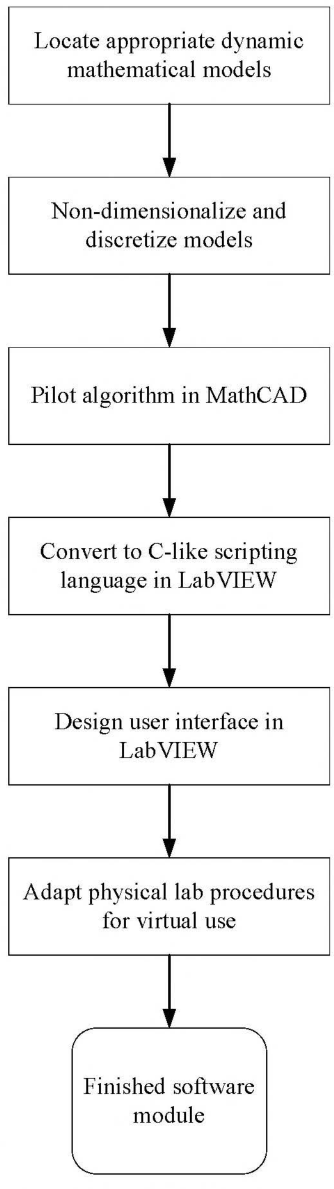

***Figure 1.** Flowchart depicting the design process for each software module of the VUOL.*

DEVELOPMENT OF THE VIRTUAL UNIT OPERATIONS LABORATORY

We sought to introduce computer-based experiments into our curriculum by simulating our senior unit operations laboratory. The unit operations laboratory at Texas Tech University is employed to reinforce to senior chemical engineering students the basic chemical engineering principles associated with various pieces of process equipment. The major pieces of equipment used in our laboratory include a double-pipe heat exchanger, a packed column ammonia absorber, and a cooling tower. The unit operations laboratory is also used to familiarize students with the safety concerns regarding each piece of equipment and about operational issues. The equipment used is comparable to pilot-scale units of industrial laboratories. We expect the students to acquire the following specific engineering knowledge and skills in the unit operations lab course: 1) implement laboratory and process plant safety; 2) analyze experimental data; 3) understand and apply the theories of shell and tube heat transfer, gas absorption, and humidification; and 4) scale equipment to the industrial level using pilot plant data. In addition to these specific chemical engineering skills, we require the students to display significant learning and development in all the areas of ABET Engineering Criterion 3 (a-k).

Guiding Principles in Virtual Laboratory Design

In our experience, effective virtual labs will have the following four key principles

(a) ***Authentic interface.*** *An important feature of a computer-based experiment is that it must be as faithful to the physical experiment as possible. Consequently, it is necessary to run the experiment at real time. This will let the student have a "feel" for the time needed for the experiment in the physical case. Moreover, the experiment should incorporate as many practical factors (that make the system deviate from ideal behavior) as possible. Another way to improve the "reality" factor is to make the indicators and controls in the computer-based experiment as similar as possible to the physical ones.*

(b) ***Pre-laboratory preparations.*** *In a conventional laboratory course, the students undergo pre-laboratory*

[1] *A CD containing the VUOL may be obtained by the reader at no cost by e-mailing Theodore Wiesner at Ted.Wiesner@ttu.edu.*

preparation before the actual laboratory session. This provides them the background information of what is happening, the working of the apparatus, and the results to be expected. The pre-laboratory preparation usually involves reading relevant sections of the textbook and/or working out problems that will give the basic idea about the experiment. For the student to get the maximum learning experience, pre-laboratory preparations are important, and hence, this aspect of the physical laboratory must be carried forward to the computer-based laboratory also.

(c) ***Documentation and formatting.*** *As for any software, computer-based laboratories must be well-documented and well-formatted. The documentation must be rich enough to guide the user from start of the experiment to the calculations and must be as user-friendly as possible. The formatting should be easy enough for the student to follow without excessive training. This helps a student to learn and practice the sessions without the need for another person for guidance.*

(d) ***Equivalent learning experience.*** *It is very easy to do mathematical computations using computers. In a computer-based experiment, however, calculations done manually in a physical experiment should be performed manually in the computer-based experiment as well. The student learns much of the theory behind the experiment by way of calculating various parameters needed. Otherwise, he or she would fail to appreciate the importance of the various parameters. Once the student is required to calculate other parameters using the available data, the relative importance of the various parameters becomes apparent. In short, the computer-based experiment must attempt to give the same learning experience that the physical lab would provide.*

GENERAL SOFTWARE PARADIGM

This section explains the general software design methodology used to create each module of the virtual unit operations laboratory. The design is a multi-step process and is illustrated in Figure 1 (previous page). The finished software module is a LabVIEW virtual instrument (.vi) file.

Locate Appropriate Dynamic Mathematical Models

The starting point for the development of each module is to find out appropriate dynamic mathematical models for the process under consideration. The models must be dynamic in order to faithfully represent the unit operation under study, particularly its unsteady state behavior.

This involves a thorough review of published models for a unit operation.

By way of example, we illustrate the mathematical treatment of a double pipe heat exchanger. The energy balances for this experiment are given in Eqs. (1) and (2).[12]

$$\frac{\partial T}{\partial t}+v\frac{\partial T}{\partial z}=\frac{4U}{\rho C_p D_1}(T_s-T) \quad \text{(tube-side)} \quad (1)$$

$$\frac{\partial T_s}{\partial t}+\operatorname{sgn} v_s\frac{\partial T_s}{\partial z}=\frac{4D_1U}{\rho_s C_{ps}\left(D_2^2-D_1^2\right)}(T-T_s) \quad \text{(shell-side)} \quad (2)$$

The exchanger is subject to the following initial and boundary conditions:

$$\begin{aligned} &T(z,0)=T_0(z)\\ &T_s(z,0)=T_{s0}(z)\\ &T(0,t)=T_{inlet}(t)\\ &T_s(0,t)=T_{s,inlet}(t) \quad \text{cocurrent}\\ &T_s(L,t)=T_{s,inlet}(t) \quad \text{countercurrent} \end{aligned} \quad (3)$$

T is the tube-side temperature, t is time, v is the tube-side velocity averaged across the cross-section, and z is the distance along the exchanger. U is the overall heat transfer coefficient, D_1 is the diameter of the inner tube, T_s is the shell-side temperature, and ρ and C_p are the density and the heat capacity of the tube-side fluid. The subscript s indicates the analogous properties of the shell-side fluid; the subscript 0 indicates initial conditions; and sgn = +1 or –1, indicating cocurrent or countercurrent flow respectively. L is the length of the exchanger.

Nondimensionalization and Discretization

We now introduce the following dimensionless variables.

$$\text{dimensionless time } \tau=\frac{tv}{L} \quad (4)$$

$$\text{dimensionless exchanger length } Z=\frac{z}{L} \quad (5)$$

$$\text{dimensionless tube-side temperature } \theta=\frac{T-T_{inlet}}{T_{s,inlet}-T_{inlet}} \quad (6)$$

$$\text{dimensionless shell-side temperature } \theta_s=\frac{T_s-T_{inlet}}{T_{s,inlet}-T_{inlet}} \quad (7)$$

The nondimensionalized energy balances are a pair of partial differential equations (PDEs).

$$\frac{\partial\theta}{\partial\tau}+\frac{\partial\theta}{\partial Z}=a\cdot(\theta_s-\theta) \quad (8)$$

$$\frac{\partial\theta_s}{\partial\tau}\pm\gamma\cdot\frac{\partial\theta_s}{\partial Z}=a_s\cdot(\theta-\theta_s) \quad (9)$$

The initial and boundary conditions become Eq. (10) in dimensionless form.

$$\begin{aligned} &\theta(Z,0)=\theta_0(Z)\\ &\theta_s(Z,0)=\theta_{s0}(Z)\\ &\theta(0,\tau)=0\\ &\theta_s(0,\tau)=1 \quad \text{cocurrent flow}\\ &\theta_s(1,\tau)=1 \quad \text{countercurrent flow} \end{aligned} \quad (10)$$

The quantities a, a_s, and γ are lumped parameters.

$$a = \frac{4U}{\rho C_p D_1} \cdot \frac{L}{v}$$

$$a_s = \frac{4 D_1 U}{\rho_s C_{ps} \left(D_2^2 - D_1^2 \right)} \cdot \frac{L}{v}$$

$$\gamma = \frac{v_s}{v} \tag{11}$$

Eqs. (8) and (9) are discretized into recurrence formulae in the time dimension to provide an open-ended simulation. Note that one could use standard, built-in solver utilities in either MathCAD or in LabVIEW to obtain the numerical solution. Both of these specify the end-times, however, and solve the system as fast as the computational platform permits. For real-time simulation of an experiment, it is necessary to run an open-ended simulation that allows user interaction with the simulator as time progresses. Hence, the need to write out and solve the discretized model *in extenso*.

We employ Lax's modification to the FTCS method (forward in time, centered in space) to numerically discretize and solve Eqs. (8)-(10). The discretized forms of the PDEs become:

$$\theta_{i,j+1} = \frac{1}{2}(1+c)\cdot\theta_{i-1,j} + \frac{1}{2}(1-c)\cdot\theta_{i+1,j} + \alpha\cdot\left(\theta_{s_{i,j}} - \theta_{i,j}\right) \tag{12}$$

$$\theta_{s_{i,j+1}} = \frac{1}{2}(1+\mathrm{sgn}\cdot c_s)\cdot\theta_{s_{i-1,j}} + \frac{1}{2}(1-\mathrm{sgn}\cdot c_s)\cdot\theta_{s_{i+1,j}} + \alpha_s\cdot\left(\theta_{i,j} - \theta_{s_{i,j}}\right) \tag{13}$$

The index i denotes time, and j denotes space. The quantities c and c_s are the Courant numbers for the two sides of the unit operation, and α and α_s are the dimensionless lumped parameters. The compositions of these four quantities are given as follows.

$$c = \frac{v\Delta t}{\Delta z} = \frac{\Delta\tau}{\Delta Z}, \quad c_s = \frac{v_s \Delta t}{\Delta z} = \frac{v_s}{v}\frac{v\Delta t / L}{\Delta z / L} = \frac{\gamma\Delta\tau}{\Delta Z}$$

$$\alpha = \frac{4U\Delta\tau}{\rho C_p D_1}\frac{L}{v} = a\Delta\tau, \quad \alpha_s = \frac{4UD_1\Delta\tau}{\rho_s C_{ps}\left(D_2^2 - D_1^2\right)}\frac{L}{v} = a_s\Delta\tau \tag{14}$$

We performed similar procedures using ammonia balances on the gas and liquid phases of the gas absorber,[14] and energy

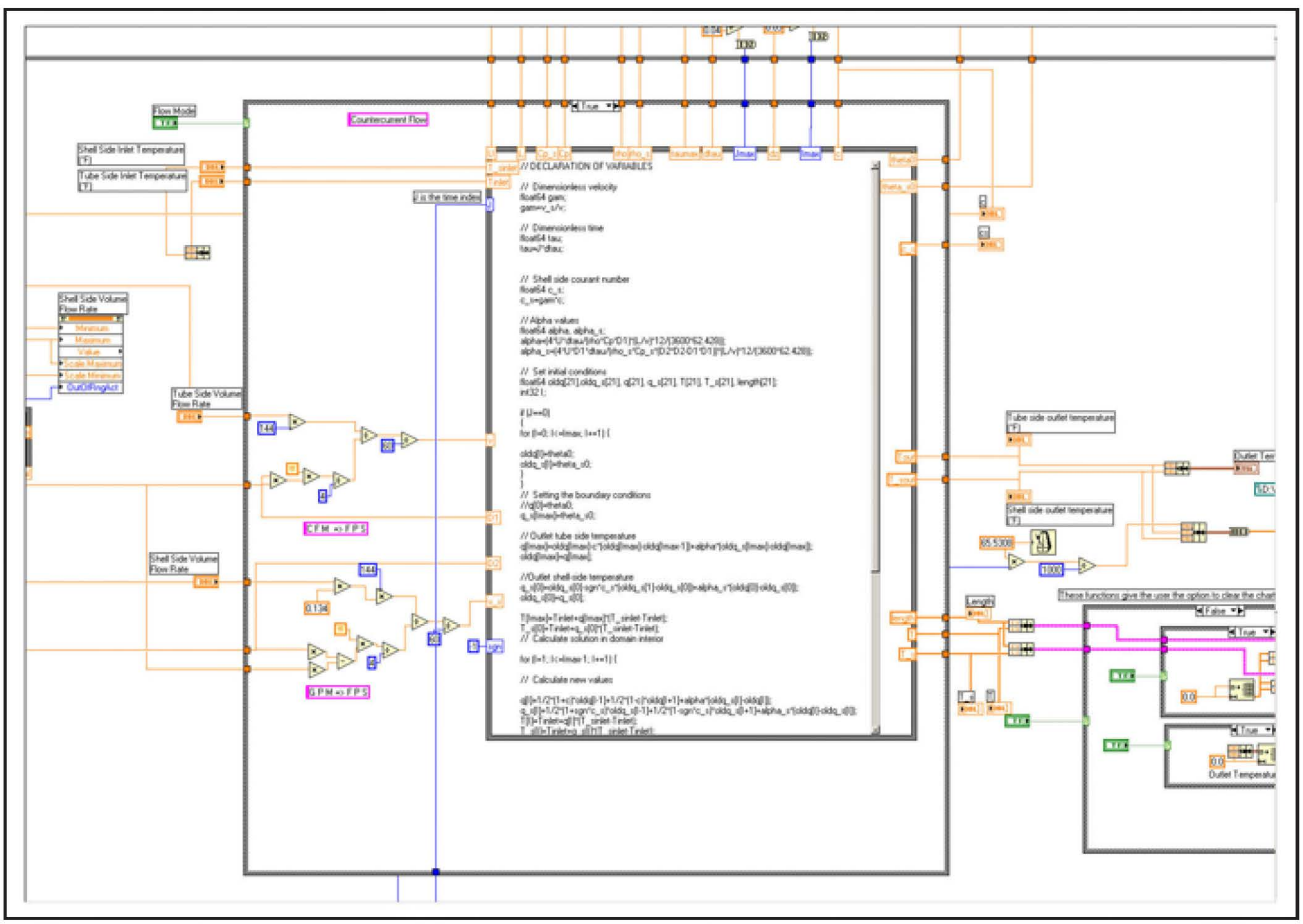

Figure 2. *Heat exchanger algorithm coded in LabVIEW (block diagram programming mode).*

balances on the air and water phases of the cooling tower.[15] Interestingly and conveniently, the dimensionless models for all three experiments have the same generic form as Eqs. (8) and (9). Thus we obtained recurrence formulae in the time dimension similar to Eqs. (12) and (13) for the absorber and cooling tower as well.

Pilot Algorithms in MathCAD

Before coding the discretized mathematical model into the distributable end product, the algorithm should be tested for stability and accuracy. Particularly, dimensional consistency must be assured when combining inputs expressed in different unit systems such as the SI and English systems. For these reasons, we piloted our algorithms in MathCAD software. In the first coding of the algorithms, MathCAD's built-in units handling was enabled, eliminating dimensional errors during translation. Because the scripting language in LabVIEW does not handle units automatically, the MathCAD pilot was then modified with automatic unit handling disabled and appropriate conversion factors introduced into the code. When the results from the code without the units were the same as the results from the code with the units, we were assured that we had the correct conversion factors.

Validate Against Physical Laboratory Data

Once suitable dynamic models have been identified, the next step is to validate the modeling results against experimental data. In our department, we reconciled our mathematical models with the results students obtained from the physical experiments at Texas Tech University in prior years.

Convert MathCAD Code to C-like Scripting Language in LabVIEW

As mentioned earlier, authenticity of the interface is an important factor in maintaining a good learning experience for the student from the computer-based lab. In the Virtual Unit Operations Laboratory, we chose to use LabVIEW as the front-end tool. Of the many factors that compelled us to choose LabVIEW, the most important was the fact that the controls and indicators provided by LabVIEW have a real "look and feel." LabVIEW comes with a palette of indicators and controls that are very much similar to the physical ones.

LabVIEW offers a C-like scripting language, in which one can program the algorithms piloted in MathCAD. Figure 2 (previous page) shows the heat exchanger algorithm as coded in the LabVIEW virtual instrument (block diagram programming mode). LabVIEW was intended to provide a virtual programmable interface to physical laboratory equipment. With the scripting feature, however, one can incorporate a mathematical model into the virtual instrument and use the flexible user interface to design a facsimile of the simulated equipment. The scripting block in Figure 2, in other words, replaces the physical device.

Design the User Interface

The physical heat exchanger and its interface for the double-pipe heat exchanger are illustrated in Figure 3. It is completely interactive. The user is able to alter the hot- and cold-water flow rates, as well as the temperatures of those streams. Another parameter that can be changed is the direction of flow, meaning the flow mode can be either countercurrent or cocurrent. These parameters were chosen as the adjustable ones because they are the same parameters that are adjustable in the physical experiment.

The interface features two graphs. One shows the outlet temperatures of both the shell and tube sides progressing with time. These values change until the system reaches a steady state; a change in an inlet temperature will affect both the outlet temperatures. The magnitude of the change is dependent on both the inlet temperatures and both the shell and tube-side flow rates. The other graph shows the temperature

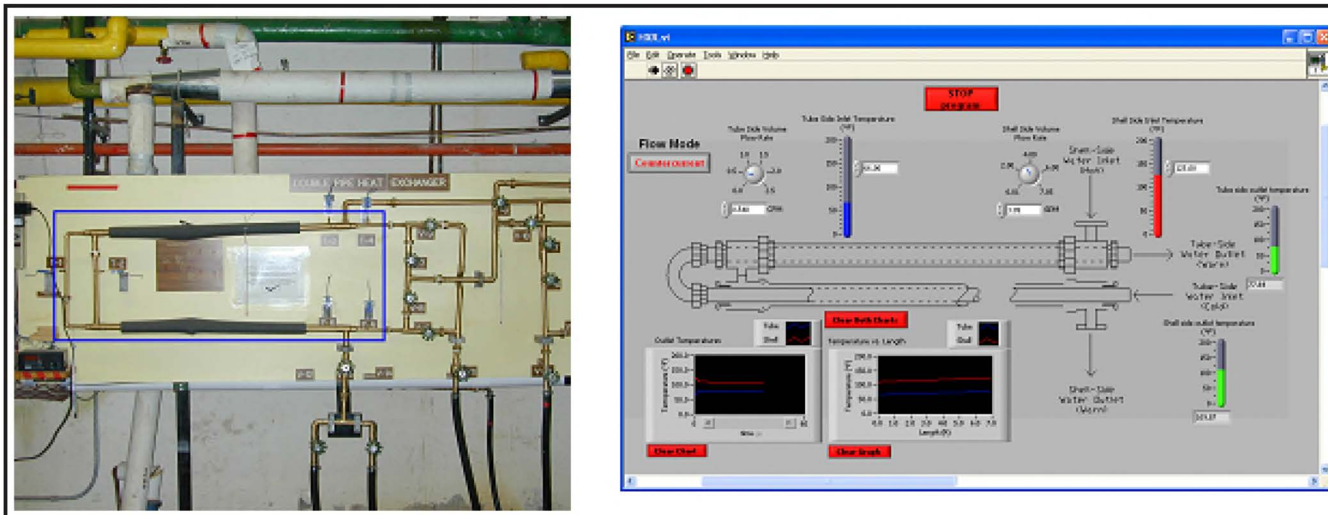

Figure 3. *The Physical Heat Exchanger and its virtual counterpart.*

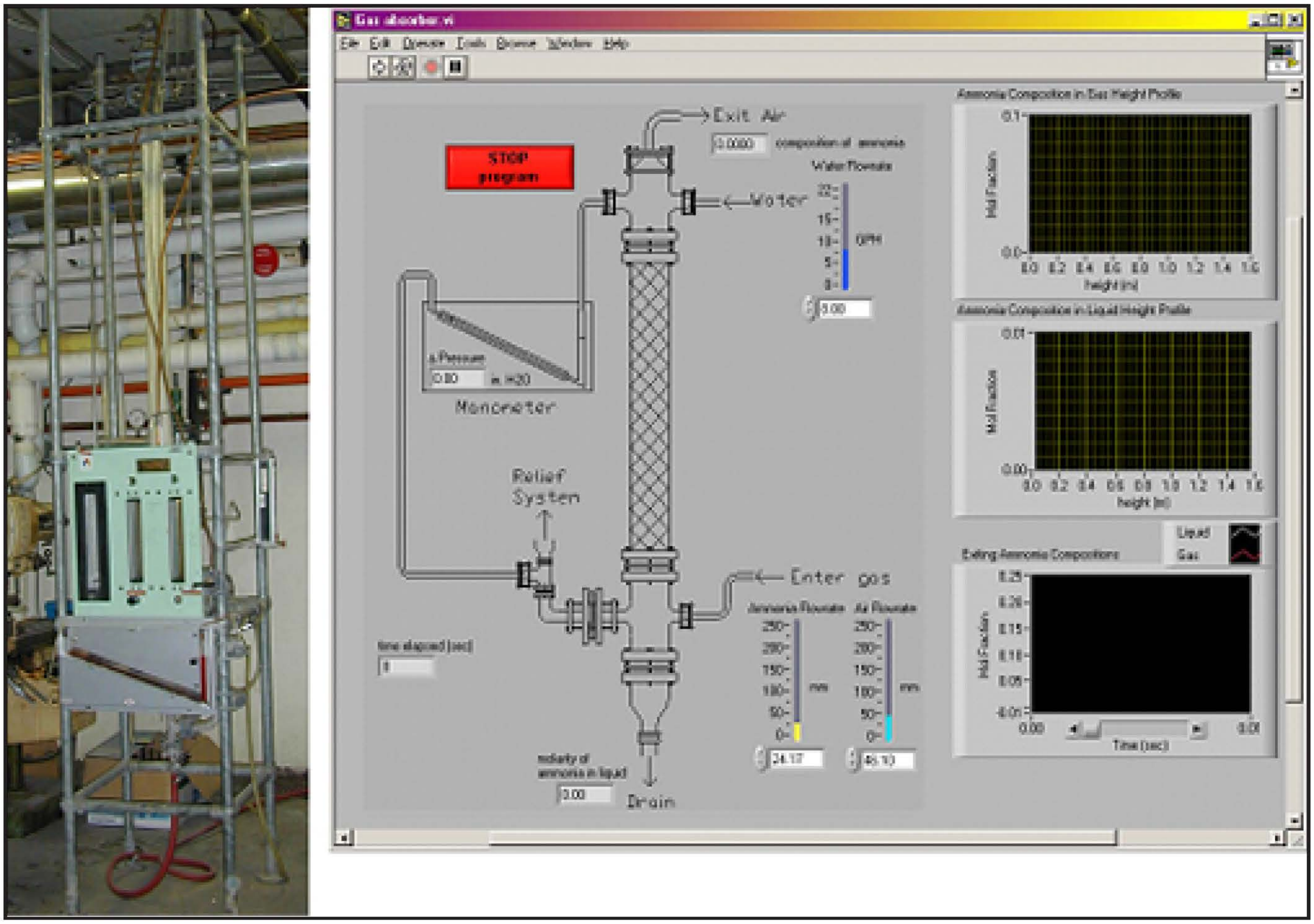

Figure 4. *Physical gas absorber (left) and the virtual gas absorber (right).*

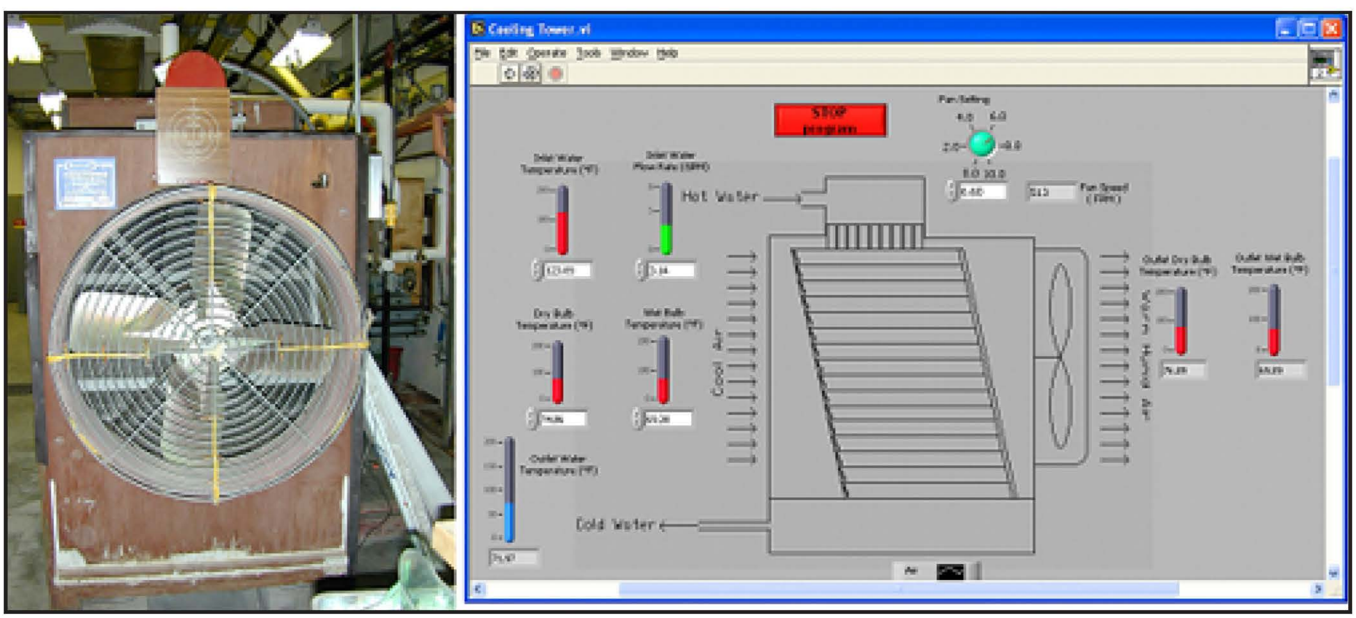

Figure 5. *The physical and virtual cooling towers.*

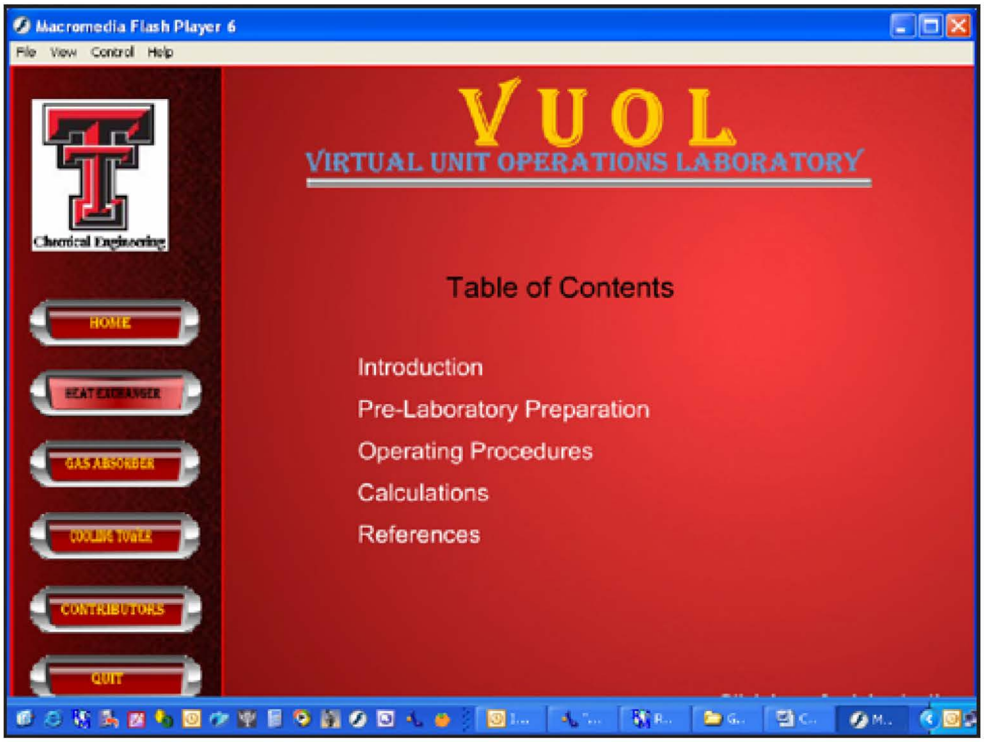

Figure 6. *The Introduction Screen to the Heat Exchanger Module.*

profile along the length of the exchanger for both tube and shell side. With a countercurrent flow mode, this graph has two parallel lines along the length of the exchanger, while in the cocurrent mode, the graph has two lines that converge toward the steady state temperatures. In both the physical and virtual versions of the heat exchanger, the students vary the shell-side flow rate, and compare the resulting Nusselt numbers with the Sieder-Tate correlation.[16] All data displayed on the graphs are also written to a text file for later analysis.

The ammonia gas absorber also has a completely interactive interface, which is a realistic model of the physical experiment (Figure 4, previous page). The user can change the flows of water, ammonia, and air as well as the inlet ammonia concentrations in the two phases. Using calibration data taken from the actual physical experiment, the dials on the interface match those of the actual rotameters found in the lab. This enables the student to emulate the actual laboratory experiment.

The gas absorber interface has seven charts. The graphs show the ammonia compositions of the liquid and gas along the height of the column, and the ammonia compositions in the exiting liquid and gas, all as a function of time. To judge the approach to steady state, the expected steady state concentration profiles in the two phases are plotted alongside the dynamic profiles (screenshot not shown). The students determine the height of a transfer unit (HTU) the number of transfer units (NTU) the overall mass transfer coefficient based upon the vapor phase ($K_y a$), and the overall column efficiency (η) as functions of the liquid to gas ratio (L/G). The simulation is based on the model of Lakshmanan and Potter.[14]

The cooling tower interface (Figure 5, previous page) is similar to the heat exchanger and the gas absorber, and is also completely interactive. The student adjusts the inlet air wet and dry bulb temperatures, fan speed, and the temperature and flow of inlet water, and observes the outlet temperatures of the water and air. Two graphs record the exit temperatures of the air and water as a function of time, as well as the development of the temperature profiles along the flow path of the water (screenshot not shown). The model displays breakthrough behavior, as is often observed in packed vessels. The students determine the heat transfer coefficient, mass transfer coefficient, and the height of a transfer unit, as a function of inlet liquid flow rate. We used the cooling tower model published by Al-Nimr[15] to simulate our cooling tower.

Integration into a Complete Suite

After choosing a model and validating it, the next step in the process of creating the virtual module is to adapt the physical lab procedures and pre-laboratory preparation to virtual use. For the VUOL, we integrated the procedures, pre-lab prep, and virtual experiments into a single self-guided computer program. An excellent platform for creating such applications is Macromedia Flash. The student follows a series of hyperlinks to move nonlinearly among the various sections of the documentation.

A welcome screen briefs the user on the purpose of the Virtual Unit Operations Laboratory. In the *Using the Product* section, the user is given basic instructions on how to use the virtual laboratory. At this point, the user must get into any one of the three modules. Again using the heat exchanger as an example, we describe the progression of performing a lesson. At the start of each module, the user is presented with a brief Introduction section regarding the experiment and the specifications of the particular equipment being modeled (Figure 6). Then, the user is taken to the Pre-Laboratory Preparation section, which details the various theoretical sections to be reviewed, and the problems to be worked out (hyperlinked) so that the student will be ready to understand and appreciate the process and outcomes of the experiment.

Next, the *Operating Procedure* for the particular virtual experiment is presented. This section includes a thorough explanation on how to start and stop the experiment, how to adjust the input parameters and how to collect and analyze the results. It also contains information regarding the various process variables and their operating limits. This is also the section in which the link to execute the LabVIEW simulation appears.

After this section comes the *Calculations* section. This section asks questions, relevant to the experiment module, that the students must answer. Also, all relevant additional information necessary to solve the problems (graphs, example figures) is provided in this section (hyperlinked). The questions are carefully chosen so that they will hone the skills of the student attempting to answer them, with regard to the experiment. Moreover, the *Calculations* section does not give information regarding how to answer the problems. This is intentional, because the *Pre-Lab Preparation* section already provided information regarding the basic theoretical knowledge to acquire and the student is expected to solve the problems in the *Calculations* section by applying that knowledge. This will help the student to develop the ability of applying general theory to a specific case. The documentation for a module ends with the *References* section, which contains citations used for the compilation of the documentation.

Compilation and CD Authoring

The next step after the completion of the Flash shell is to combine the modules and the documentation into one unit that is easy to distribute. To do so, we compiled a CD containing the modules and the documentation. Also, for increasing the product's ease of use, the CD has the "autorun" feature, which brings up a message window providing the user a choice of two options: 1) Run the VUOL directly from the CD, or 2) install the VUOL and support files to the hard drive. The CD-ROM was distributed in August 2004 to 156 departments of chemical engineering in the United States.

DISCUSSION

The task of developing the computer-based experiments is not trivial. First, creating a computer-based lab is a time-consuming process. The physical experiments under consideration must be modeled mathematically. The modeling requires careful analysis and study of the basic working of the process to be correctly simulated. Once the ideal situation is modeled, the external factors that are prevalent in the practical case must be identified and should be included in the computer-based experiment. This analysis takes time and effort, but it helps create an authentic tool that will familiarize the student with the practical situations. Once a model that is close enough to the physical case is obtained, the next step is to realize the model using computing tools. This again needs a careful scrutiny of the available tools to choose the best one. Altogether, to create a computer-based version of a physical lab, a large amount of background work is needed.

Second, creating a computer-based experiment requires skilled labor. In order to prepare the mathematical model of the experiment, a person competent in the experiment being modeled is needed. To transform the mathematical model into a computer-based program, a person who is competent in computer programming and who can "read" the mathematical model is required. In only a few cases are these skills found together in one person.

In replacing physical experiments with computer-based analogs, the following question arises: How much is student learning compromised by the reduction of tactile or "hands-on" learning? In fall 2002, we conducted a rigorous comparison between control groups performing physical versions of the experiments and groups performing the simulated experiments, the results of which are published in Reference 17. We found no significant difference in learning between the two delivery modes. The differential impact was measured by performance on a comprehensive exam, by student feedback on how well the objectives of ABET Criterion 3 (a-k) were met, and by student recommendations in oral presentations. The results indicate that student learning is not adversely affected by the partial replacement of physical experiments with computer-based laboratory exercises.

The tactile engineering laboratory should remain an integral part of the engineering curriculum. Students gain confidence from turning real valves and seeing real results in a real lab. Real processes exhibit myriad unanticipated effects and random influences, which students learn a great deal from encountering. It is possible to program in some nonidealities, but it is difficult for the course designer to anticipate all possible effects and to simulate them faithfully.

Nonetheless, in view of the increasing use of computers in the chemical process industries, the instructional material can and should be adapted to the increasing use of information technology in the manufacturing industries. It is important that virtual experiments be devised that retain high fidelity to their physical analogs. For example, if the laboratory pedagogy requires students to write their own experimental procedures for physical equipment, students conducting virtual experiments should also have to write experimental procedures. The software should be written to allow this activity. From our experience in the development, implementation, and assessment of a virtual unit operations laboratory, we have concluded that a considerable up-front time investment is required. This investment is justified, in most cases, by the enhanced reliability and flexibility of the lessons in conjunction with reduced instructional costs. Given the many benefits of computer-based instruction and the prevalence of computer tools in engineering practice, those designing engineering curriculums should seriously consider the use of computer-simulated experiments as an adjunct to their laboratory instruction.

ACKNOWLEDGMENT

This work was supported by a Special Grant in the Chemical Sciences from the Camille and Henry Dreyfus Foundation; fund number SG-01-090.

REFERENCES

1. Anido, L., M. Llamas, and M.J. Fernandez, "Labware for the Internet," *Computer Applications in Engineering Education*, **8**(3-4) 201 (2000)
2. Hansen, E., "The Role of Interactive Video Technology in Higher Education: Case Study and Proposed Framework," *Education Technology*, p. 13, September (1990)
3. Cooper, D., and D. Dougherty, "Enhancing Process Control Education with the Control Station Training Simulator," *Computer Applications in Engineering Education*, **7**(4) 203 (1999)
4. Murphy, T., V.G. Gomes, and J.A. Romagnoli, "Facilitating Process Control Teaching and Learning in a Virtual Laboratory Environment," *Computer Applications in Engineering Education*, **10**(2) 79 (2002)
5. Kadiyala, M., and B. Crynes, "A Review of Literature on Effectiveness of Use of Information Technology in Education," *J. Eng. Ed.*, **89**(2) 177 (2000)
6. Powell, R.M., H. Anderson, J. Van der Spiegel, and D.P. Pope, "Using Web-Based Technology in Laboratory Instruction to Reduce Costs," *Computer Applications in Engineering Education*, **10**(4) 204 (2002)
7. Sorby, S.A., G. Walker, M. Yano, V. Glozman, K. Kochersberger, J. Mathers, J. McKinney, J. Schulman, and M. Young, "Modernization of the Mechanical Engineering Curriculum and Guidelines for Computer-aided Engineering Instruction," *Computer Applications in Engineering Education*, **7**(4) 252 (1999)
8. TTU Chemical Engineering Advisory Board. personal communication
9. Hale, J.C., and H.L. Sellars, "Historical Data Recording for Process Computers," *Chemical Engineering Progress*, **77**(11) 38 (1981)
10. Anderson, M., and P. Whitcomb, "How to Document Process-Monitoring-And-Control Systems," *Chemical Engineering*, **88**(18) 117 (1981)
11. Basta, N., G. Ondrey, and S. Moore, "The Latest Miniature Computers Let Engineers Access Data, on the Spot," *Chemical Engineering*, **102**(8) 33 (1995)
12. Schiesser, W., and C.A. Silebi, *Computational Transport Phenomena*, Cambridge University Press, Cambridge, UK (1997)
13. Pozrikidis, C., *Numerical Computation in Science and Engineering*, Oxford University Press, New York (1998)
14. Lakshmanan, C.C., and O.E. Potter, "Dynamic Simulation of Packed-Type and Tray-Type Absorbers," *Industrial & Engineering Chemistry Research*, **28**(9) 1397 (1989)
15. Al Nimr, M.A., "Dynamic Thermal Behavior of Cooling Towers," *Energy Conversion and Management*, **39**(7) 631 (1998)
16. McCabe, W. L., J. C. Smith, and P. Harriott, *Unit Operations of Chemical Engineering*, 6th Ed., McGraw-Hill, Boston (2001)
17. Wiesner, T., and W. Lan, "Comparison of Student Learning in Physical and Simulated Unit Operations Experiments," *J. Eng. Ed.*, **93**(3) 195 (2004) ❒