SimzLab - Interactive simulations of physical systems for active individual and team learning

by

Richard K. Herz
Chemical Engineering Program and Nanoengineering Department,
University of California, San Diego <herz@ucsd.edu>

and

Gregory E. Ogden
Chemical and Environmental Engineering Department,
University of Arizona, Tucson <gogden@email.arizona.edu>

Teaching by lecture and textbook alone does not satisfy students' needs. Many physical systems are too complex to be conveyed fully by the static plots and drawings in books. Essential to learning is active practice and application of new knowledge. Real experiments are wonderful - but cost and space constraints limit the number which can be implemented - usually from zero to a few in most courses. Interactive software simulations can engage students actively in the learning process and help them to understand and work with complex systems. Interactive simulations engage the student[2,3]. They are interesting and fun to use[4-6], and help students take responsibility for their education[7].

This paper describes a software application – SimzLab[8] - and what we have learned from developing and using it. Our main objective has been to provide students with virtual lab modules to supplement lecture courses on chemical processes.

SimzLab can distribute multiple sets of modules or "Labs" over the Internet, with each Lab hosted on its own server. The current Labs are Reactor Lab (simulations of chemical reactors) and PureWaterLab (modules on water purification and use). Programming has been done at UCSD and the explanatory text in PureWaterLab has been written at the University of Arizona.

SimzLab is a desktop application which communicates over the Internet with servers. Since it is a desktop application rather than a web browser, it has the full capability of an application such as running compiled C executables for compute-intensive simulations. Since it can connect to the Internet, it has access to protocols such as HTTP, FTP, and TCP/IP sockets.

The software has been developed using a tool with a graphical layout editor and a high-level scripting language[1]. On each client, there is one executable file which has been compiled for that platform (hardware and operating system) and which contains minimal bootstrap code or "script." All other scripts are contained in files which are cross-platform. Whenever a student goes on-line, SimzLab updates any file - both script and content - on the client for which there is a newer version on the server. Currently we only provide Mac and Windows versions of SimzLab, although Linux versions are possible.

Students download the SimzLab application from the web site at www.SimzLab.com at no cost. A few modules are provided in the initial distribution. When a student connects SimzLab to the Internet, the student can download additional modules. Once downloaded, a module can be used off-line.

Fig. 1 shows the opening window and the two courses available.

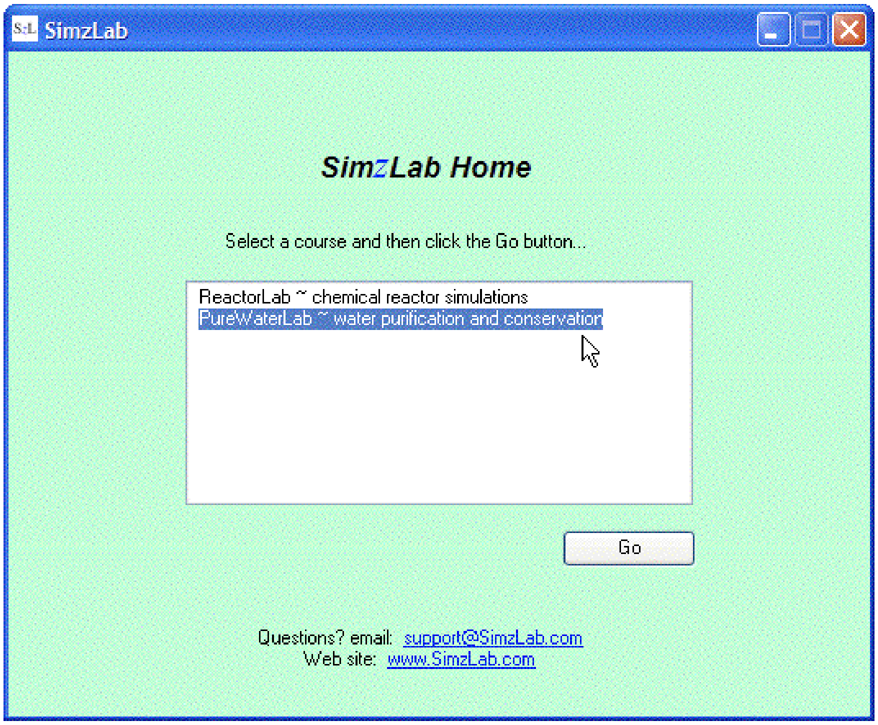

Fig. 1. Opening window showing the sets of course modules currently available.

In the Reactor Lab, many of the modules have two sections: the main entry point in which all inputs and outputs are known to the student, and a quiz section. In the quiz section, students must run experiments, analyze their data, and then check to see if they have an answer within an acceptable range. The Lab charges virtual $ for each experiment to teach students that they should not run experiments indiscriminately, and then awards them virtual $ for correct answers. The Budget Report records the history of each quiz. Students must turn in their data, analysis work, and a copy of the Budget Report which contains an authentication code. These quizzes incorporate the features Pavia[9] lists that should be exhibited by a laboratory simulation.

It is interesting to observe a new group of students start to use the software in a computer lab. Doing homework in the Lab is much different that working the usual end-of-chapter homework problems, where usually the necessary and sufficient set of data are given such that a unique answer can be obtained. Some students get the idea and use the Lab enthusiastically. Some students stare at the computer with a concerned look, unsure where to begin. They aren't given data; they have to perform experiments and take their own data and their first experiments may be under conditions which do not provide useful data.

One student was angry, protesting "We are only undergrads! We only know how to work textbook problems!" To which we thought, "Yes, that's exactly the problem!"

As far as functionality, we observed that students highly value software responsiveness (speed) and the ability to use software without having to read instructions.

The Reactor Lab has been used by students all over the world and has been translated by volunteer students and professors into Spanish and Portuguese. Although gratifying, we felt that greater use of the Lab could be obtained if more complete modules with explanatory text were provided in addition to lab simulations. And so, in PureWaterLab, we are developing both explanatory text with graphics and interactive simulations.

The current PureWaterLab course includes an overview on how ultrapure water (UPW) is produced, and detailed modules on specific process components including reverse osmosis, UV photo oxidation, and ion exchange. Fig. 2 shows the Directory window of PureWaterLab.

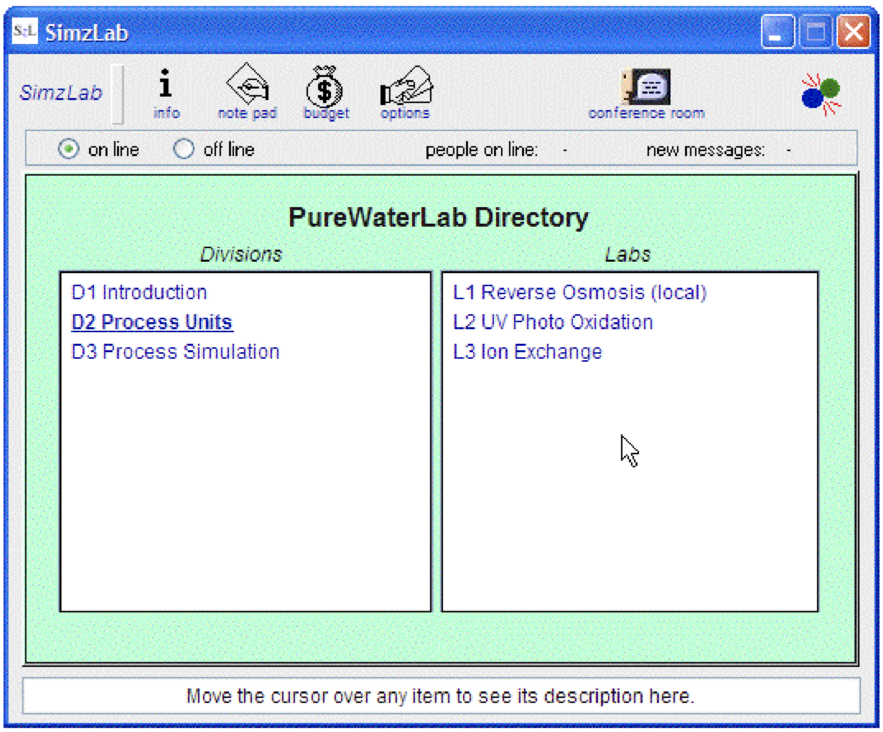

Fig. 2. Directory showing divisions and lab modules in PureWaterLab.

A fourth module in the Process Units division, activated carbon treatment, is under development. The modules include background information, design parameters, treatment capabilities and uses as well as interactive quizzes and homework problems. The simulations are designed to maximize flexibility from "black box" canned response with little user input except for inlet conditions to detailed inputs and operating conditions for more advanced users. Fig. 3 shows a typical the text explanation window and the basic level simulation window open. The module presented in Fig. 3 describes reverse osmosis.

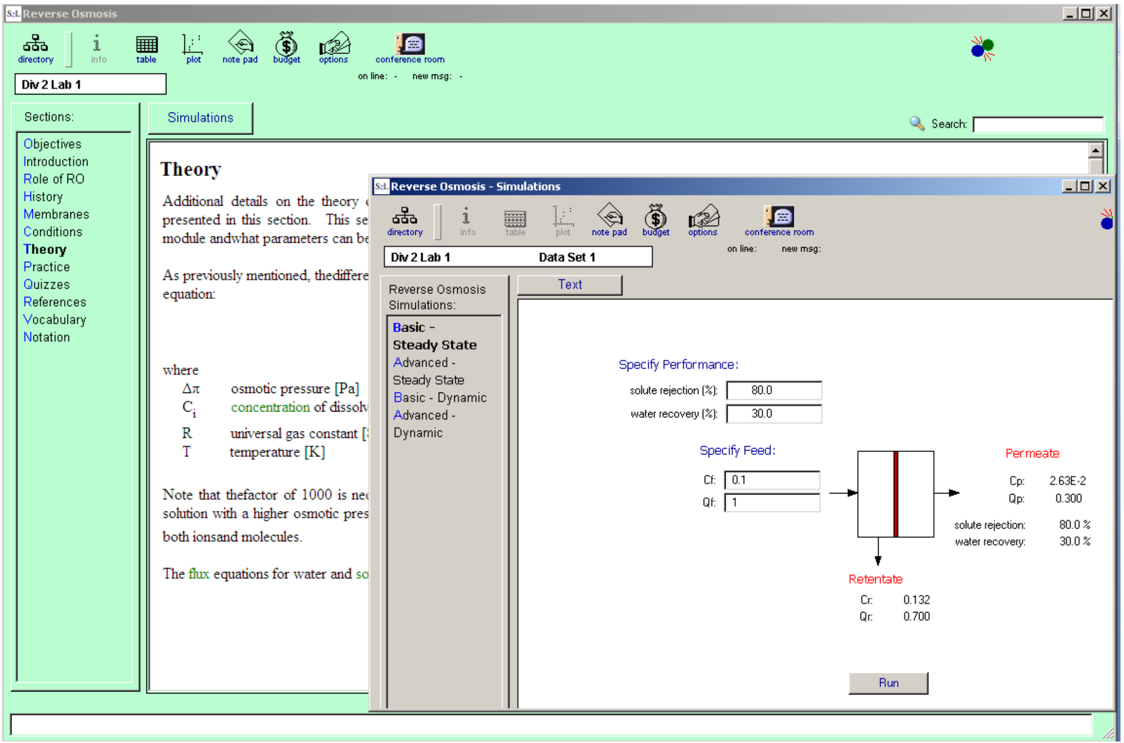

Fig. 3. Explanation window and simulation window for Reverse Osmosis module.

A more advanced simulation for reverse osmosis is presented in Fig. 4.

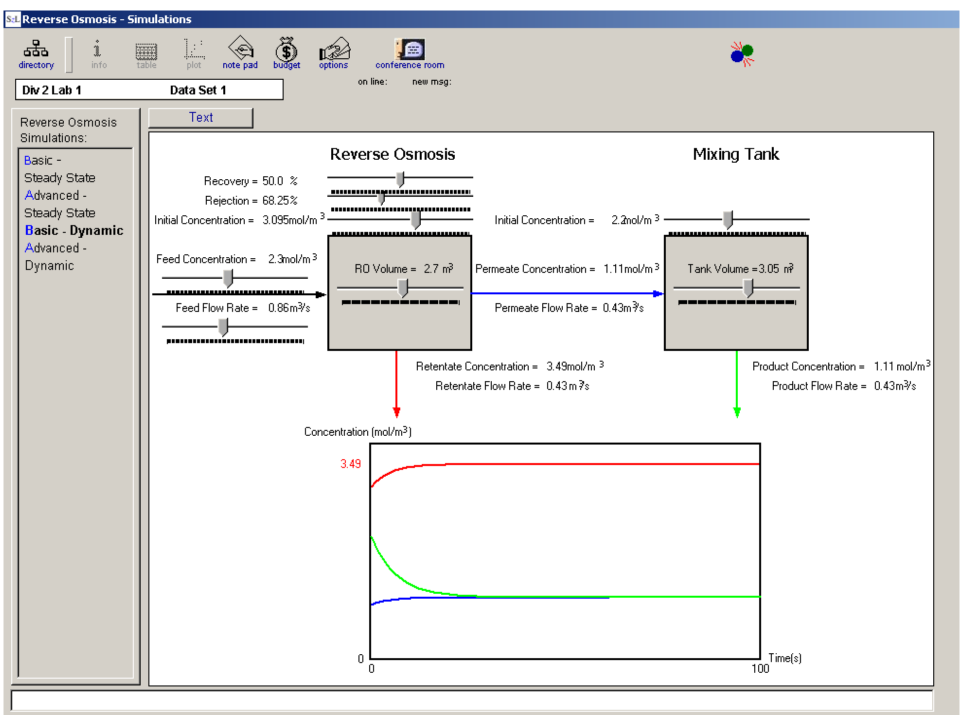

Fig. 4. Screen shot of Reverse Osmosis Dynamic Simulation.

At the start of preparing the text and graphics, we discovered that the team members at the University of Arizona wanted more control over formatting than could be provided by the development tool. Therefore, we switched to letting the authors create web pages. The web pages are rendered by the default browser on the client (Internet Explorer on Windows and Safari on Mac) in windows that are "skinned" by PureWaterLab. Any web component that can be rendered by Internet Explorer or Safari can be displayed by SimzLab.

This raises the question, "Why not distribute the complete modules as web pages accessed with a standard web browser?" The reason is that we wouldn't be able to deliver simulations with the full range of functionality that we require. In addition, SimzLab can add additional functionality not present in standard browsers. An example is that it scans pages for words in the Glossary, adds HTML tags next to the first occurrence of each word found, and displays definitions in an information field below the web page as the use mouses over the now highlighted vocabulary words.

One feature we would like to develop further is the "Conference Room." This is somewhat of a cross between a bulletin board and a chat room. It gets most active the night before a homework assignment is due at UCSD, when students ask each other for help. On one occasion, we had a three-way conversation between a professor in San Diego, a postdoctoral associate from Turkey in Ann Arbor, and an undergraduate student in Turkey.

A recent addition to PureWaterLab is a module for constructing and conducting dynamic simulations of water purification plants. The connection to the Internet allows students at different universities to collaborate on running the same plant, with one team of students operating one part of a plant and teams at other universities operating other parts of the plant.

Students can construct a process plant by adding process units and pipes to a flowsheet, as shown in Fig. 5. Material flows between units are represented in the software by messages. In Fig. 5, the blue unit on the flowsheet is the local proxy of the actual unit, which is located on a remote computer. Messages between computers are written in XML text and sent over Internet between the computers via TCP/IP sockets. Since messages between computers are in platform-independent XML text, the simulators on individual computers can be written in any computer language running on any hardware. Since the “internals” of units do not have to be known or “exposed” to users, future uses may include units posted by companies to allow students and prospective customers to experiment including a commercial unit in their process. The potential advantages of distributed dynamic simulators have been discussed by Shemeikka, et al[10].

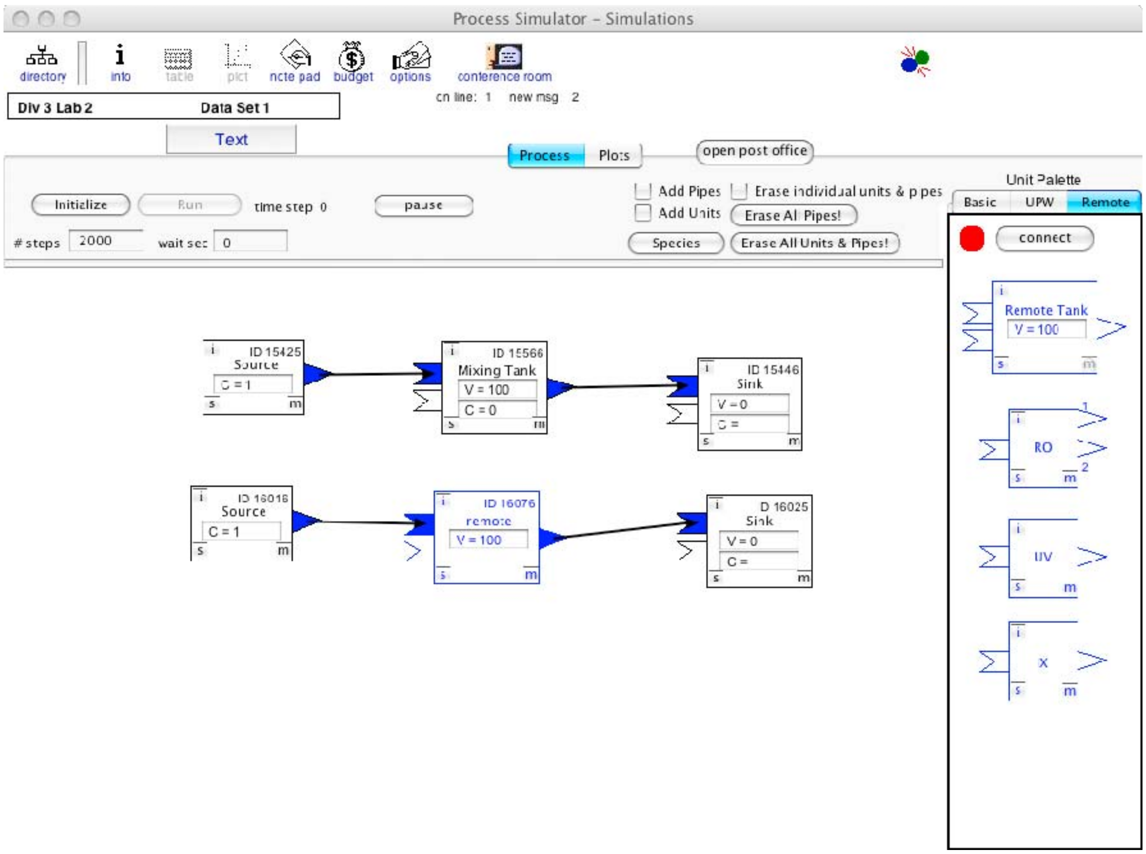

Fig. 5. Distributed dynamic process simulator in PureWaterLab.

Acknowledgements

Development of SimzLab has been supported by the U.S. National Science Foundation through awards including DUE-0443044. Students contributing to the project include UCSD graduate student Hsu-Wen Hsiao and UA undergraduate students Erica Reiner, Chris Poole, Megan Meyer, Jason Kim, Adam Burkhart, David Bartz, and Michael Hwang.

References

1. "Revolution" software development tool, developed and distributed by Runtime Revolution Ltd., Edinburgh, Scotland <http://www.RunRev.com/>, accessed February 19, 2009.

2. Zacharia, Z. (2003) Beliefs, attitudes, and intentions of science teachers regarding the educational use of computer simulations and inquiry-based experiments in physics, *Journal of Research in Science Teaching,* vol. 40, p. 792.

3. Chou, C. (1998) The effectiveness of using multimedia computer simulations coupled with social constructivist pedagogy in a college introductory physics classroom, Doctoral dissertation, Teachers College, Columbia Univ., New York.

4. Richmond, B., Peterson, S. & Vescuso, P. (1987) An Academic User's Guide to STELLA. High Performance Systems, Inc.: Hanover, NH.

5. Geban, O., Askar, P., & Ozkan, I. (1992) Effects of computer simulations and problem solving approaches on high school students, *Journal of Educational Research*, vol. 86, p. 5.

6. Hounshell, P.B. & Hill, S.R. (1989) The microcomputer and achievement and attitudes in high school biology. *Journal of Research in Science Teaching*, vol. 26, p. 543.

7. Grayson, D. (1996) Use of the computer for research on instruction and student understanding in physics. *American Journal of Physics*, 55, 108–119.

8. SimzLab, available at <http://www.SimzLab.com>, accessed February 19, 2009.

9. Pavia, D. L. (1990) SQUALOR: The design of a laboratory simulation, *Academic Computing*, February, p. 8.

10. Shemeikka, J., Klobut, K., Heikkinen, J., Sipilä, K., & Rämä, M. (2007) First Experiences with Coupled Dynamic Simulation of Building Energy Systems and the District Heating Network. *Proceedings of Clima 2007 WellBeing Indoors*.

Biographical Information

Richard Herz, Ph.D. Associate Professor, University of California, San Diego. Dr. Herz teaches and conducts research in the areas of chemical reaction engineering, heterogeneous catalysis, and chemical process simulation and design.

Greg Ogden, Ph.D., P.E.   Associate Research Professor, University of Arizona. Dr. Ogden is a registered professional engineer (Chemical) with over 20 years of industrial engineering experience. He has worked on numerous engineering processes in the energy, nuclear, environmental, and aerospace arenas.   Dr. Ogden has brought his industrial experience to the classroom in both community college and university settings. He taught environmental science, chemistry and engineering programs at Pima Community College in Tucson for over 10 years and is currently coordinating the freshman engineering design course for the College of Engineering at the University of Arizona in addition to managing the undergraduate laboratory experiences for the Department of Chemical Engineering.