NSF Award DUE-0443044 - Final Report, July 2010

PureWaterLab - Conservation Education and Research Through Interactive Simulation

This attachment to the final report discusses the accomplishments of the project. The goal of the project is enhanced student learning about physical systems through interaction with software simulations. We strongly believe that interactive simulations should be supplied with course materials in addition to conventional static text, static images, and pre-recorded videos that play the same way each time.

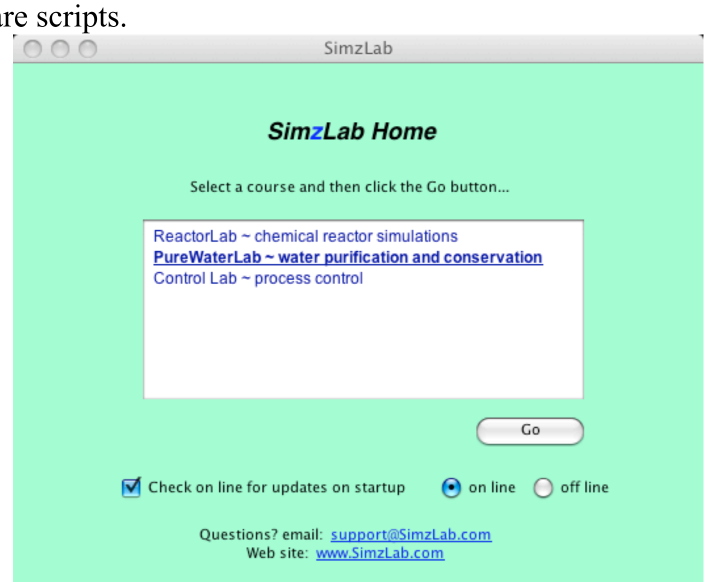
During 2009-2010, the last year of the project, we continued work in a no-cost extension. Accomplishments for the year include development of additional modules for the existing ReactorLab and PureWaterLab sets of modules, and development of a new set of modules for process control.

To date, a total of 78 interactive simulations and explanatory text modules have been developed. Of this total, 31 are new interactive simulations and text modules that were developed under this award.

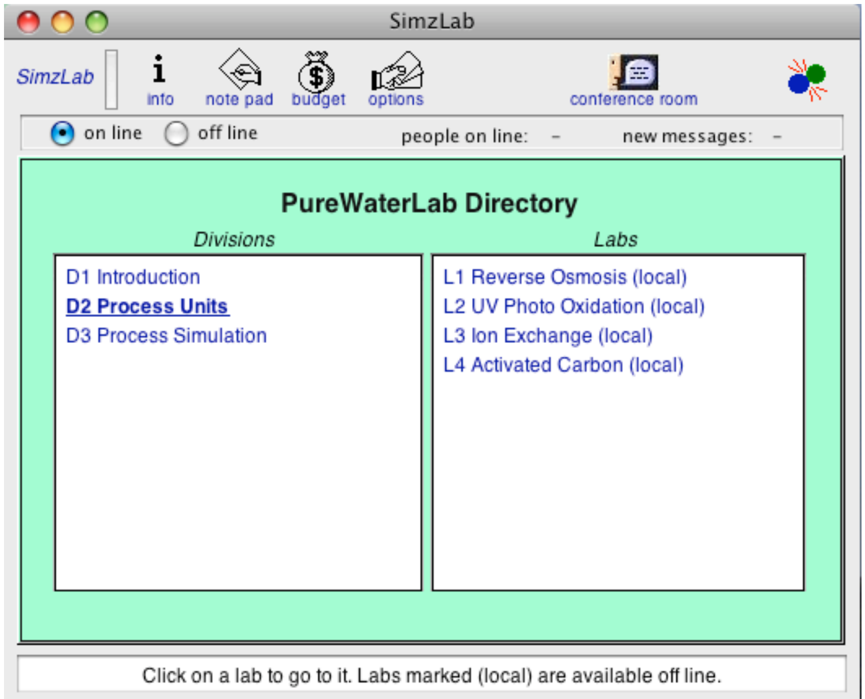
In prior work in this project and a prior DUE-CCLI project, we developed two sets of interactive simulation modules: ReactorLab and PureWaterLab. The software codes for these two projects were similar and developing them separately and keeping them both current was a headache. Therefore, we developed a common code-base and a common software application for distribution of both sets of modules. We call this common delivery vehicle, SimzLab. The software can be downloaded at no cost from SimzLab.com. During the last year of the project, a new set of modules or a "course," Control Lab, was developed.

SimzLab is a desktop application that is integrated with the Internet and associated software on web servers. In the current web jargon, the Lab is a "rich Internet application." When on line, a student can access new modules and communicate in the Conference Room with other students. Updates to software are automatically downloaded and installed. When off line, the student can continue to work on the modules they previously accessed while on line.

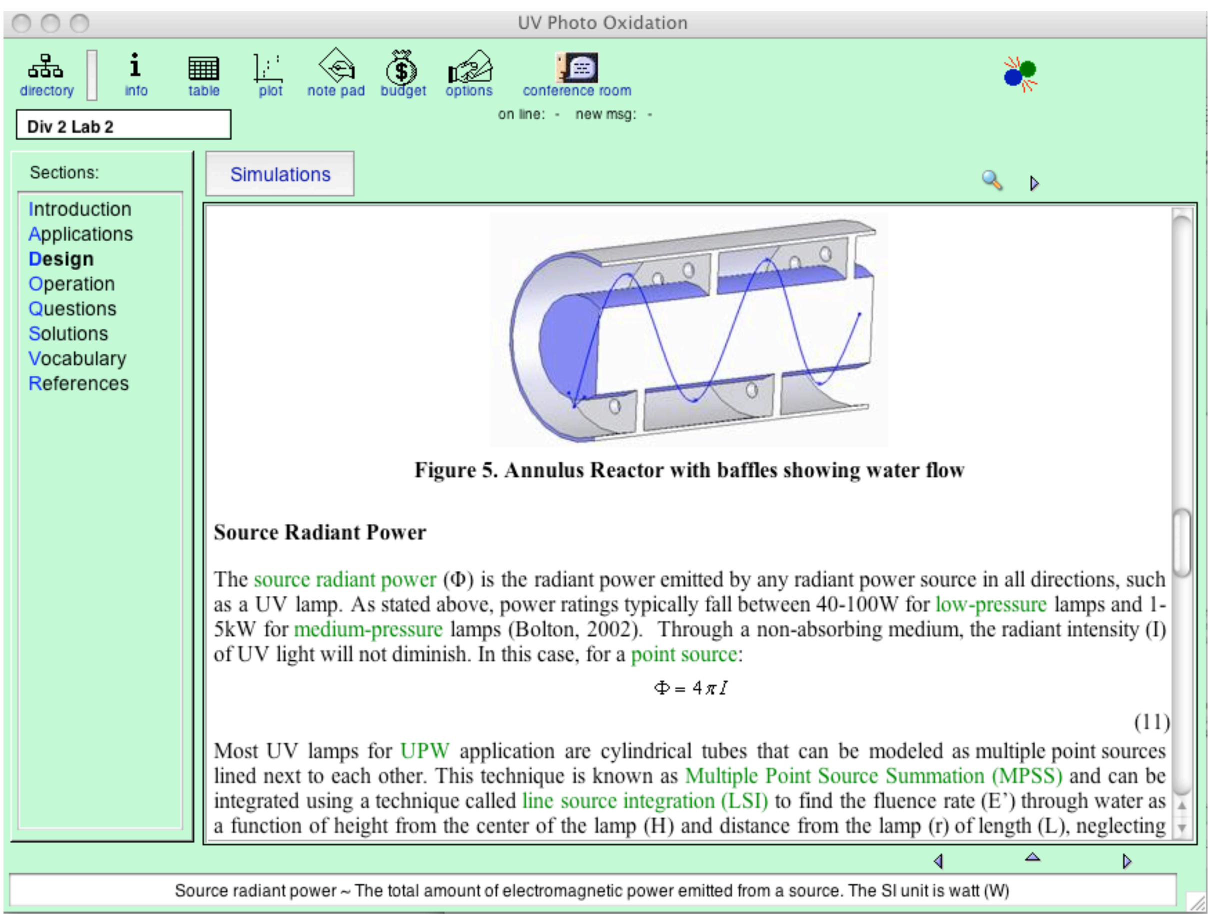
The figure below is the home screen showing the courses currently available. A course can be located on a different server than any of the other courses, and different than that of the SimzLab server, which delivers updates of the core software scripts.



This is the Directory screen of PureWaterLab. The modules or "Labs" are arranged in "Divisions" (screens are not shown at same scale here).



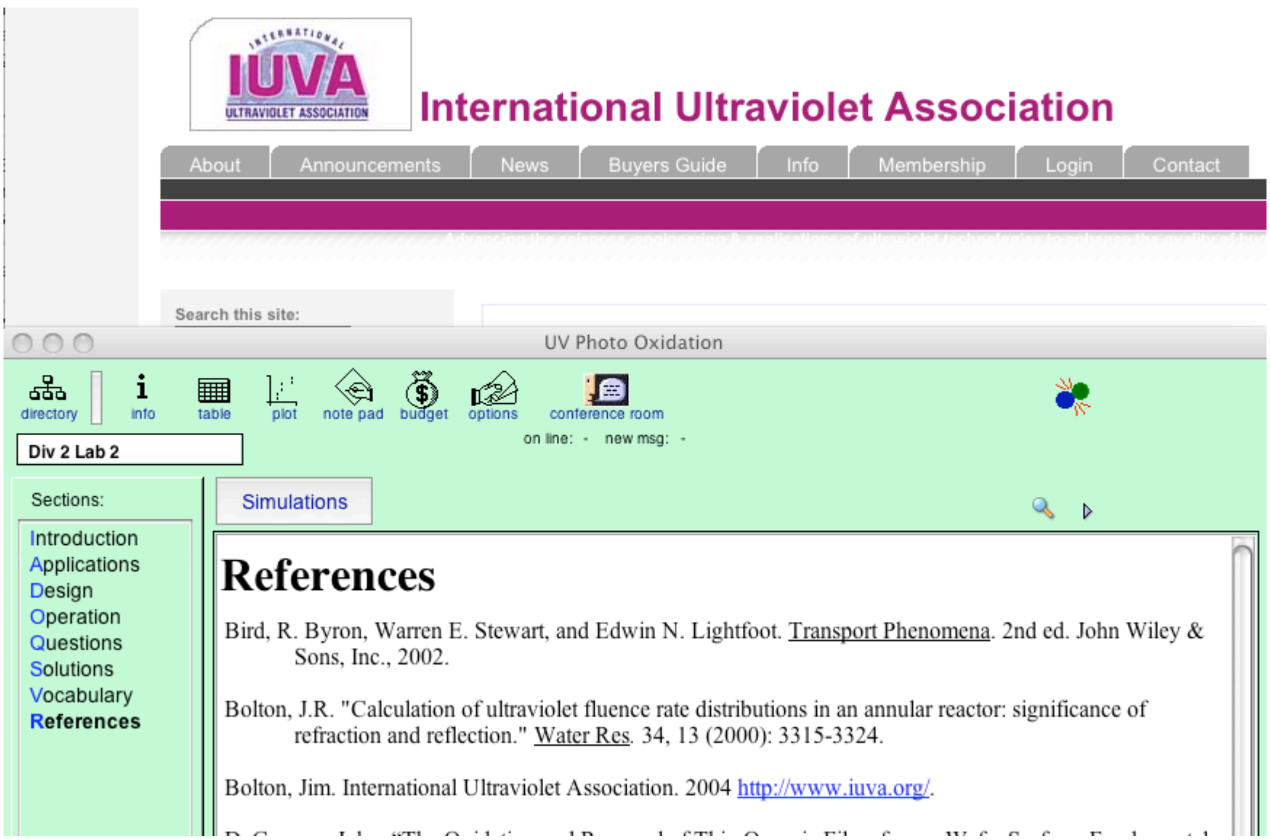
This screen shows the explanatory part of a module on the UV Photo Oxidation method of water purification.



This project was a collaboration. The University of California, San Diego (UCSD) part of the team worked on the software programming and the interactive simulations. The University of Arizona (UA) part of the team worked on the main module content, including text, graphics, math equations, and assessment components.

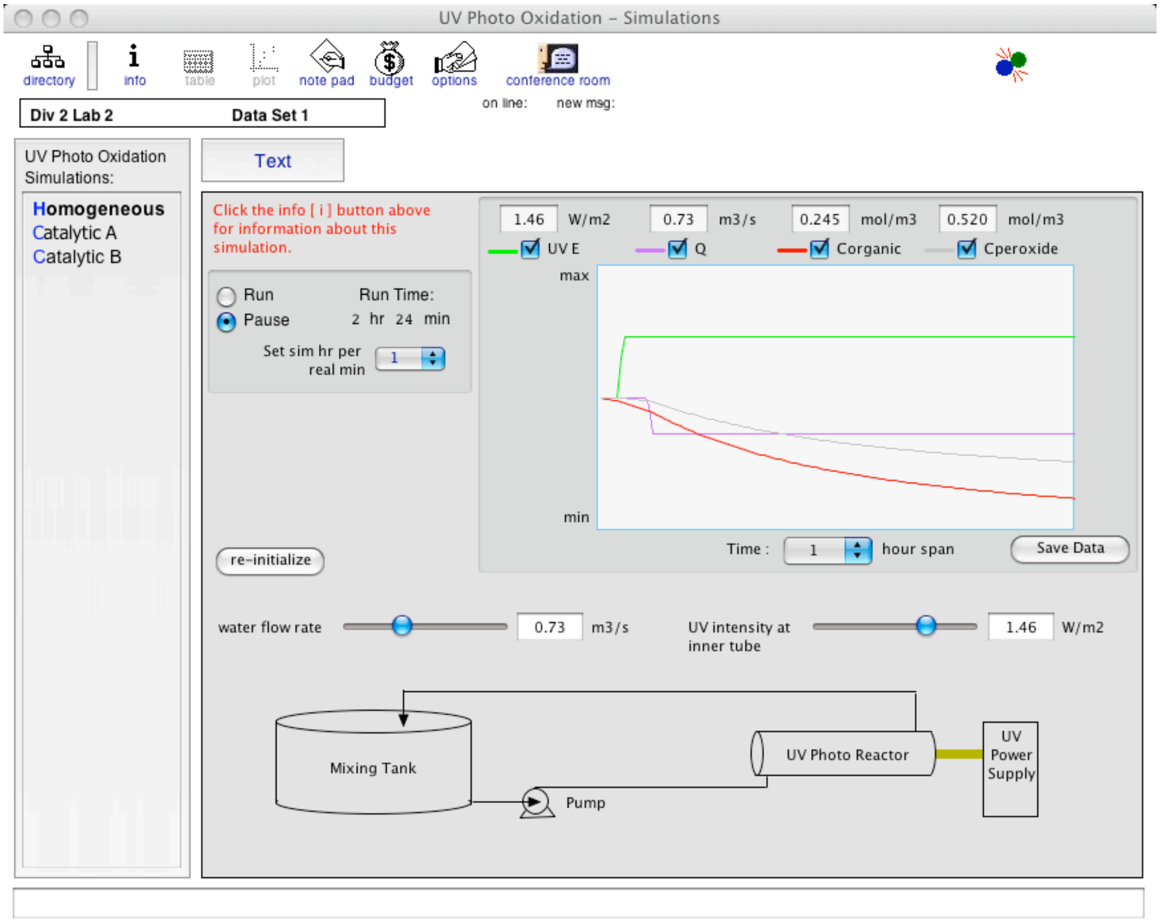
A work process was developed such that the UA group could develop and add new and revised content easily without having to involve the software group at UCSD. The UA group develops content as standard web pages and uploads them to the PureWaterLab server. Whenever a student is using PureWaterLab on-line, the software automatically detects new and updated modules and downloads them for on- or off-line use.

Text can contain links to web sites external to PureWaterLab. These links are opened in the user's web browser.

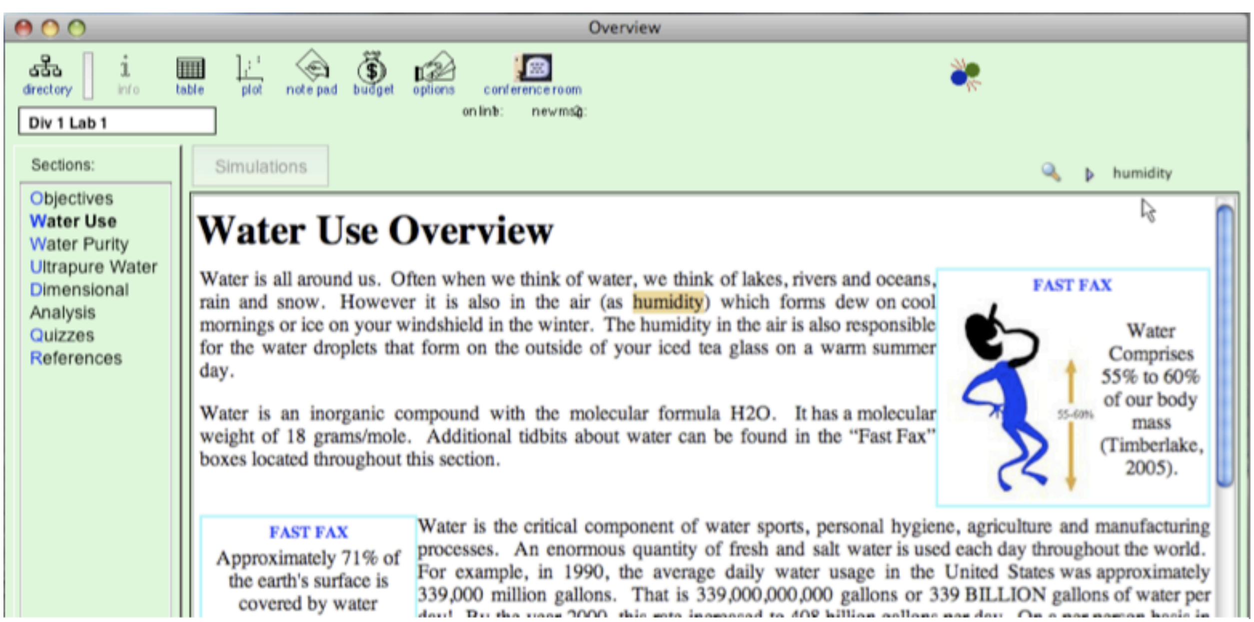


The advantage of displaying web pages in PureWaterLab, as compared with a standard web browser, is that many other features are additional available in PureWaterLab, such as the interactive simulations.

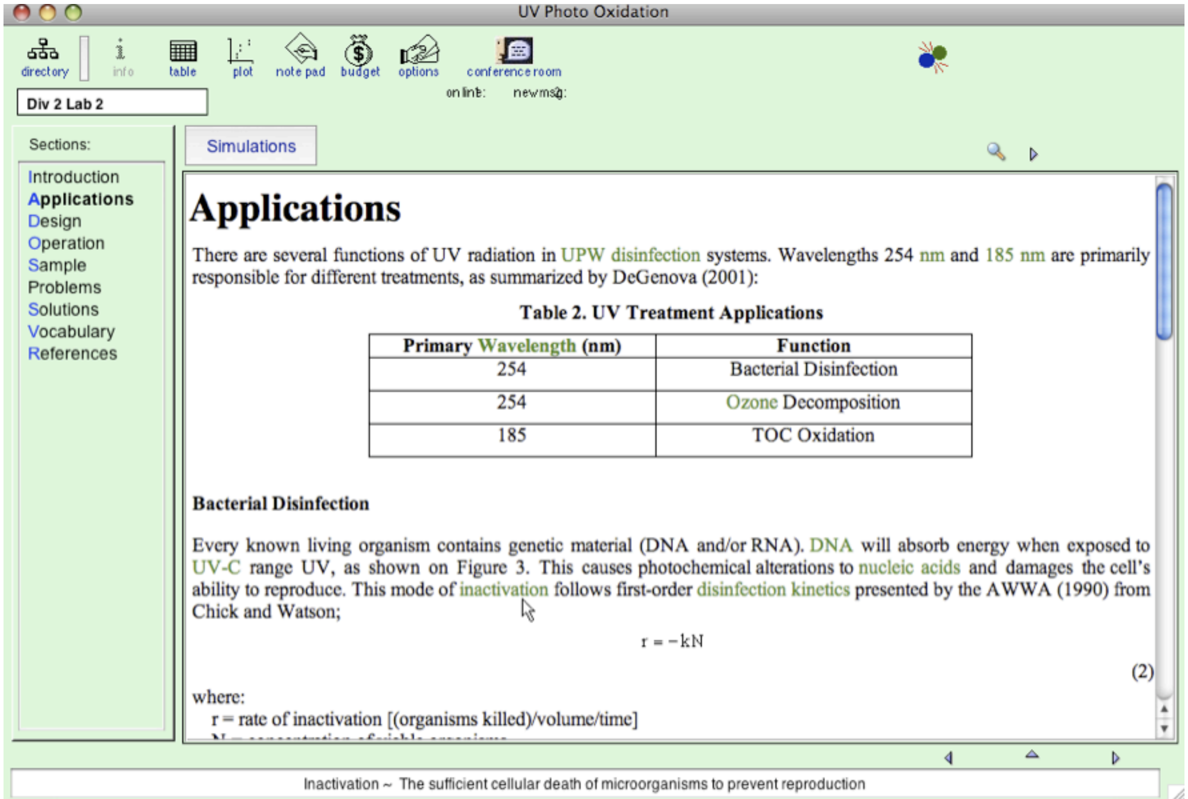
A simulation in the UV Photo Oxidation module is shown below. The student can vary parameters such as water flow rate and UV light intensity and see how the system responds. Students can save data to disk files and analyze the data. For example, in this UV simulation, students could analyze the data in order to determine values of rate coefficients.



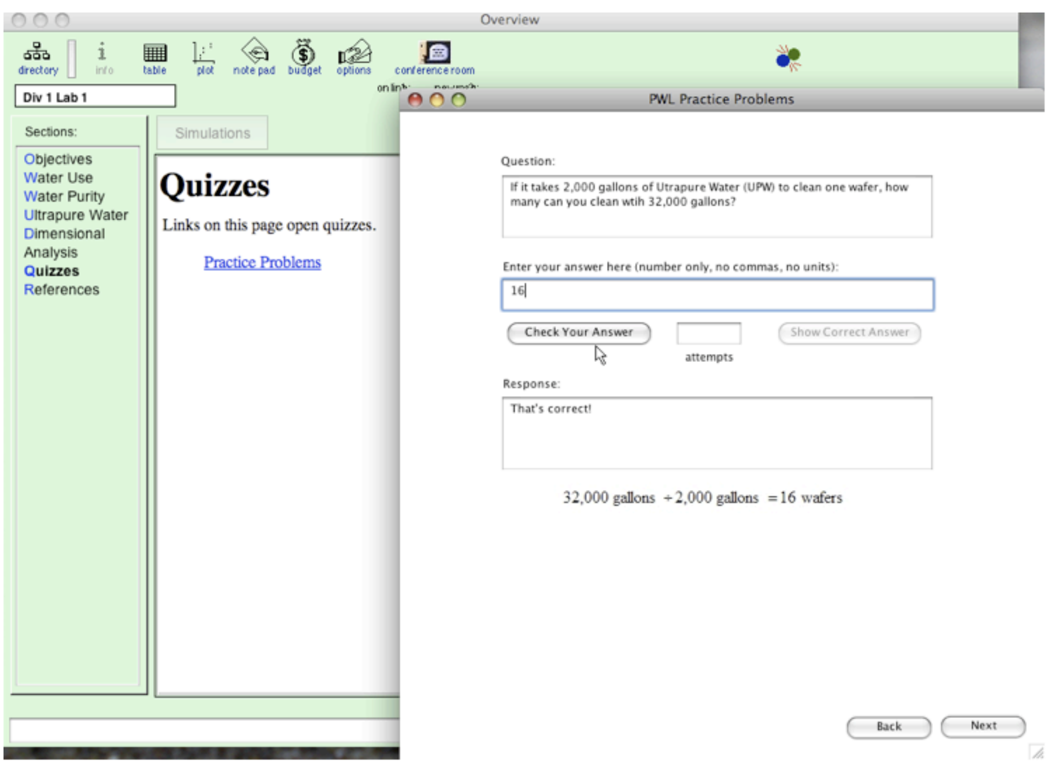
Several ways were developed to help students search text and understand vocabulary. A result of a search for a word is shown here.



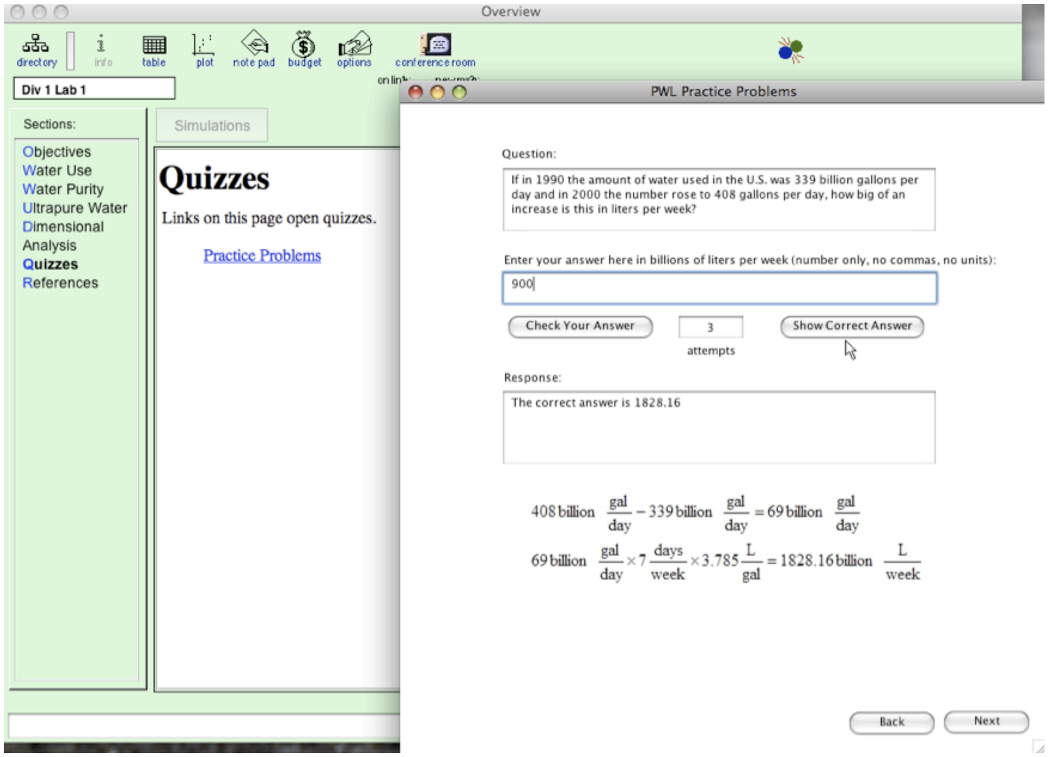
One special feature provided by the software is automatic scanning of text for words listed in the vocabulary section. Other than preparing the vocabulary section, the content authors do not have to do anything else. The software highlights vocabulary words automatically, and the definition is shown at the bottom of the window when the student passes the cursor over a highlighted word, as shown below.



Another new feature being added are quizzes to assess student learning. Several different types of questions are available, and the software automatically scores the answers and provides feedback, as shown here.



If a student enters incorrect answers several times, they are allowed to view the correct answer.

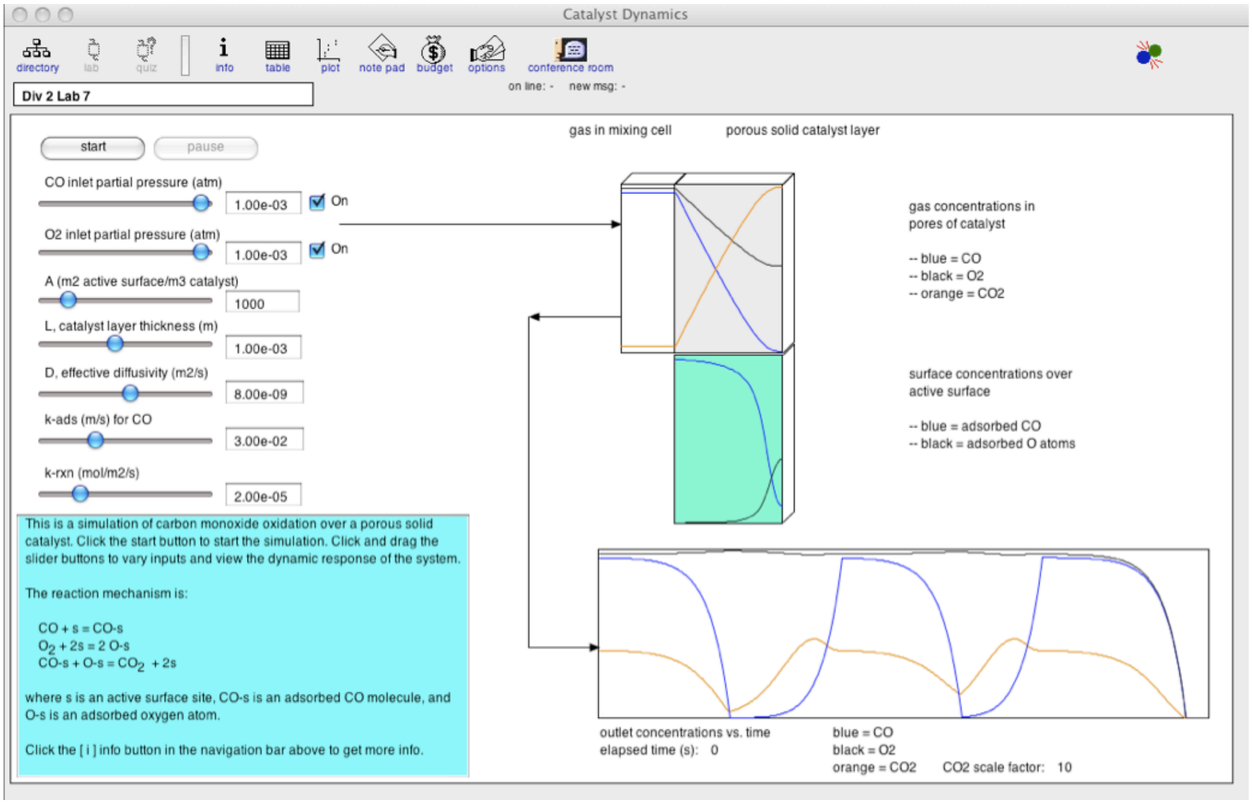


A special type of interactive simulation that provides very detailed, compute-intensive simulations of physical processes was developed. Whereas the cross-platform software language that is used for most of the project has sufficient speed to handle most simulations, compiled executables need to be used for more demanding simulations.

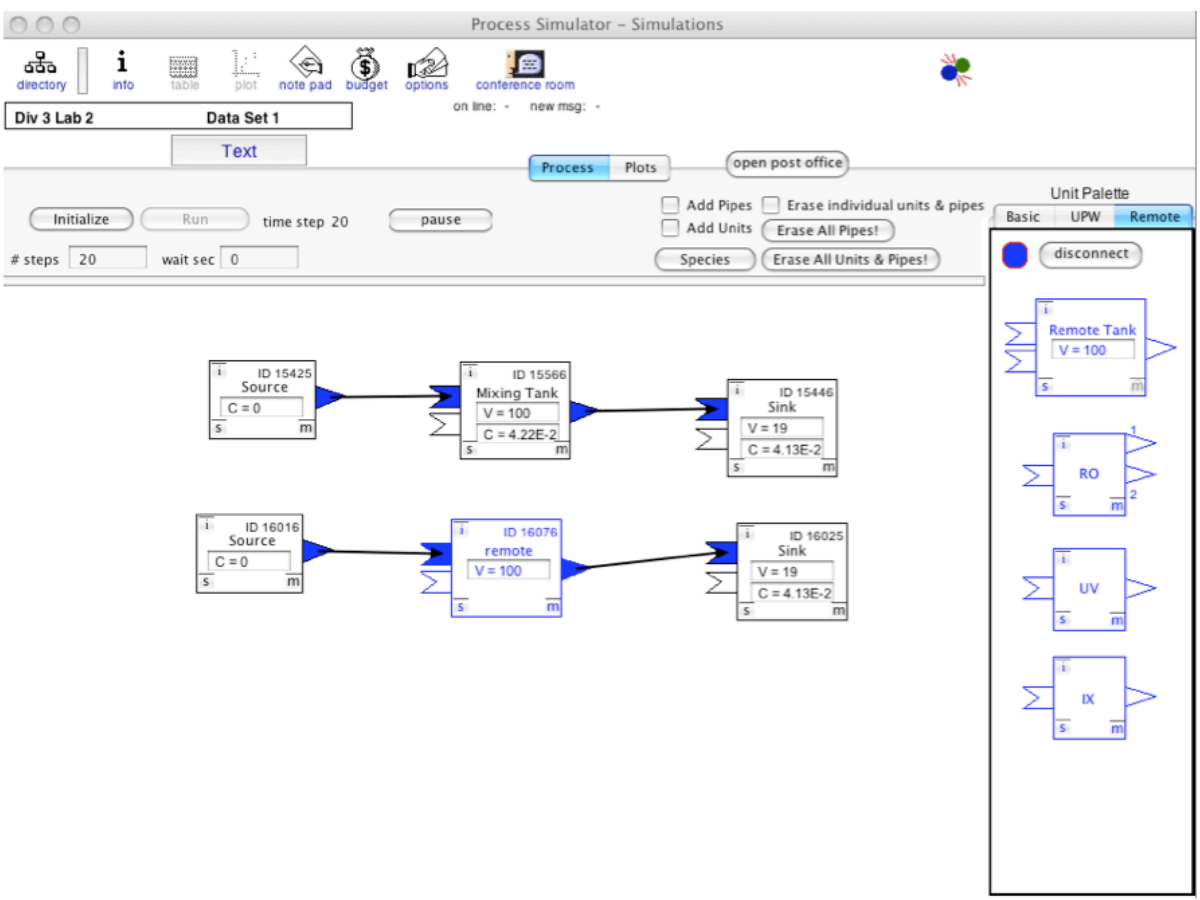
One of our visions is to provide undergraduate students with research-grade simulations where appropriate. Our development this year in this area was to couple simulations in C++ compiled executable files to the Graphical User Interface or GUI and also to deliver those compiled files along with the other module components.

The figure on the next page shows a dynamic simulation in Reactor Lab of the CO oxidation reaction in a porous solid catalyst. When running, the concentration profiles in the gas mixing cell, the pores and surface of the catalyst layer, and the outlet gas concentrations continuously update. The student can change parameters and the gas inputs by moving the sliders.

At each time step in the GUI, the module sends the current values of the inputs to the compiled executable file which then integrates the system equations for many internal time steps, returns the output to the GUI, and then the GUI updates the plots. This procedure will be used to develop other detailed simulations.



Progress is continuing to be made on the plant simulator to allow for inter-campus collaboration on simulations. The figure below shows the simulator with two processes in parallel.



Each process consists of a water source, a pipe to a mixing tank, and a pipe to a sink. There is a contaminant component in the water leaving the source, whose concentration "C" is shown.

In the top process, all units are "local" in the sense that they are computed on the client computer on which this copy of PureWaterLab is resident. In the bottom process, the blue unit is a local proxy for the actual unit which resides on a remote computer. When running, the water flowing through the pipes is carried by "messages" which are passed from unit (software object) to unit. Messages sent to a local proxy for a remote unit are sent over the Internet to the remote unit via TCP/IP socket messages. The remote unit updates its state and sends a return message back to its proxy on the original computer.

Messages sent between local units are written in the language of the local client. Messages sent across the Internet are written in cross-platform XML text. Because these messages are written in XML, the different simulators on the different computers which participate in a simulation can be written in any computer language and run on any operating system on any hardware.

A major accomplishment was to speed up the execution of the simulation, and to test running collaborative simulations between UCSD and U. Arizona.

Development of the simulator will continue. Our vision is to enable student groups at different universities to collaborate on designing and running plants. During a collaborative simulation, students will be able to communicate via text messages using the technology developed for the Lab's current Conference Room.

SimzLab software, including PureWaterLab, is used by students around the world. Presented here are results from an analysis of the download logs and server logs for the first few months of 2010.

Activity from San Diego, California IP addresses is NOT COUNTED below. Thus, these statistics do not include the PI or his students.

For the period March 26 to June 14, 2010, the desktop application files (apps) were downloaded by 657 unique users (unique IP addresses) from 67 countries. Desktop apps includes SimzLab and the Spanish and Portuguese versions of Reactor Lab.

Top 20 countries for download of desktop app, March 26-June 14, 2010:
(number after % is number unique users)

IR [Iran] % 99
US [United States] % 70
IN [India] % 64
BR [Brazil] % 60
ID [Indonesia] % 45
MX [Mexico] % 36
CO [Colombia] % 18
PH [Philippines] % 15
MY [Malaysia] % 14
SA [Saudi Arabia] % 11
CA [Canada] % 10
ES [Spain] % 10
ZA [South Africa] % 10
AR [Argentina] % 9
EG [Egypt] % 9

KR [Korea-KR] % 9
PK [Pakistan] % 9
SD [Sudan] % 9
EU [EU] % 8
GB [United Kingdom] % 8
IT [Italy] % 8

For the period January 1 to June 16, 2010, desktop apps of 1560 unique users (unique IP addresses) from 72 countries accessed lab module files on the servers. This number (1560) of unique users is larger than the number above (657) because of students who had downloaded the desktop apps before the period above and were continuing to use the software.

Top 20 countries for accessing lab module files on server by desktop app, January 1-June 16, 2010:
(number after % is number of files)

US [United States] % 9451
CO [Colombia] % 3095
IN [India] % 2793
SA [Saudi Arabia] % 2458
EU [EU] % 1424
IR [Iran] % 1410
PH [Philippines] % 1044
IT [Italy] % 991
CN [China] % 988
CA [Canada] % 809
TR [Turkey] % 768
BR [Brazil] % 727
ZA [South Africa] % 702
KR [Korea-KR] % 507
PL [Poland] % 493
MX [Mexico] % 481
GB [United Kingdom] % 465
RO [Romania] % 445
MY [Malaysia] % 409
GR [Greece] % 363