

Purdue University

Purdue e-Pubs

---

Department of Computer Science Technical  
Reports

Department of Computer Science

---

1997

## Softlab: A Virtual Laboratory Framework for Computational Science

A. C. Catlin

M. G. Gaitatzes

Elias N. Houstis

*Purdue University*, [enh@cs.purdue.edu](mailto:enh@cs.purdue.edu)

Z. Ma

S. Markus

*See next page for additional authors*

Report Number:

97-014

---

Catlin, A. C.; Gaitatzes, M. G.; Houstis, Elias N.; Ma, Z.; Markus, S.; Wang, Nien-Hwa; and Weerawarana, S., "Softlab: A Virtual Laboratory Framework for Computational Science" (1997). *Department of Computer Science Technical Reports*. Paper 1351.  
<https://docs.lib.purdue.edu/cstech/1351>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries.  
Please contact [epubs@purdue.edu](mailto:epubs@purdue.edu) for additional information.

---

**Authors**

A. C. Catlin, M. G. Gaitatzes, Elias N. Houstis, Z. Ma, S. Markus, Nien-Hwa Wang, and S. Weerawarana

**SOFTLAB: A VIRTUAL LABORATORY  
FRAMEWORK FOR COMPUTATIONAL SCIENCE**

**A.C. Catlin  
M.G. Gaitatzes  
E.N. Houstis  
Z. Ma  
S. Markus  
J.R. Rice  
Nien-Hwa Wang  
S. Weerawarana**

**Department of Computer Sciences  
Purdue University  
West Lafayette, IN 47907**

**CSD-TR 97-014  
February 1997**



*SoftLab : A virtual  
laboratory framework for  
computational science*

---

A.C. Catlin, M.G. Gaitatzes, E.N. Houstis, Z. Ma  
S. Markus, J.R. Rice, Nien-Hwa Wang and S.  
Weerawarana

---

The process of prototyping is part of every scientific inquiry, product design, and learning activity. The new economic realities require the rapid prototyping of manufactured artifacts and rapid solutions to problems with numerous interrelated elements. As part of the SoftLab project, we are investigating the issues involved in the design and implementation of a virtual laboratory that simulates and controls the functionality of a wet/dry prototyping laboratory. The central characteristics of the 21st century prototyping laboratory environments will include: i) integrated view of the design, simulation, manufacturing, and training phases of the underlying prototyping process, ii) virtual environments capable of controlling both the experimental (physical) and computational prototypes together with their interactions, iii) knowledge based systems for handling the increasing amount of knowledge and data a researcher must synthesize, compare, trade-off, and optimize, iv) 3-D numerical simulation systems that support the different possible models of the underlying physical process, v) virtual computational environments that exploit HPC and multimedia technologies towards the goal of eliminating or minimizing the need for physical prototypes and construction aids. Creating a prototyping laboratory with such design goals presents a challenge in itself as well as a number of research opportunities. In this paper we present a software framework for a virtual laboratory and report on our experiences in designing and implementing two software prototyping laboratories, microelectronics material (MicroSoftLab) and bioseparation (BioSoftLab) chemical engineering laboratories.

## 1.0 Introduction to SoftLab

---

The fields of computational science and computational engineering force us to address the challenge of harnessing high-performance computers and exploiting them through very high level systems that target a certain class of problems. Such systems require a wide range of expertise plus a flexible and diverse array of equipment. The SoftLab framework should provide the infrastructure and facilities that serve the needs for basic research broadly focused on building problem-solving environments (PSE) for applied, experimental work for computational science. The facilities include:

- High-performance graphics processors to support scientific visualization, geometric modeling and design, and multimedia graphical user interfaces for parallel programming and programming in the large.
- High-performance computers.
- Software that allows application researchers to make full use of these facilities comprising a rich spectrum of high-performance workstations, powerful parallel machines, and dedicated graphics processors.
- Virtual laboratories that provide scientists with a full range of functionality, both for experimental activities and for the mathematical modeling and simulation of the experimental processes.

The laboratory presents an ideal environment in which to accept and meet the challenges of computational science and engineering. Issues that must be addressed include mathematical software, electronic prototyping, geometric modeling, parallel algorithms, databases, software engineering, and computer systems. This facility is a catalyst for significant research in computational science and engineering, both within the Computer Science Department, as well as in other departments. It accelerates the maturation of a number of collaborative, interdisciplinary research projects that are beginning to create a critical mass of applications-oriented research and expertise in computational science. This effort is *SoftLab*. [1]

A central group effort within SoftLab addresses the full range of issues in the future paradigm of computational science. These include collaboration with researchers in chemical engineering, mechanical engineering, biology, and biochemistry. In this paper, we describe the design and implementation of two prototype virtual laboratories, BioSoftLab and MicroSoftLab. Each is a hybrid problem-solving environment which supports the seamless and intelligent interaction of the experimental and computational models in a single, unified environment. We will show how these specialized environments attempt to fulfill the SoftLab research objective.

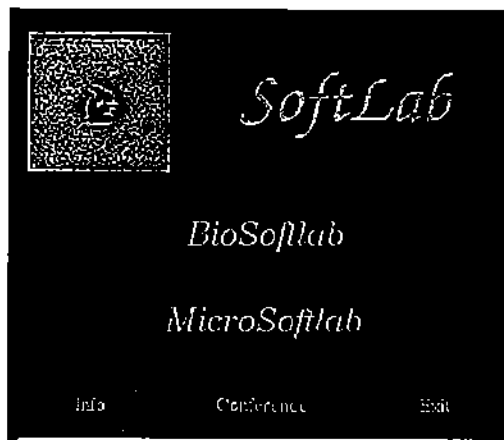
## 2.0 The SoftLab Virtual Laboratories

---

At the highest level, SoftLab acts as a gateway to the virtual laboratories that have been implemented using the SoftLab design philosophy. Each SoftLab laboratory is a complete problem-solving environment for a specific area in Chemical Engineering.

FIGURE 1.

The SoftLab gateway



The virtual laboratory is a hybrid PSE that encompasses all research activities occurring in the scientific laboratory environment, both “wet” and “dry”. It is a software layer above the experimental and computational processes which exist in the physical laboratory side by side. But the virtual laboratory attempts to bring these two models of scientific research together in a way that allows them to interact with each other, so that feedback from one can enhance and improve the methodologies applied in the other.

This software layer should support interaction with remote instruments for control of the experimental process. It must also support simulation of the experimental process - which means that it must be able to access and execute the PDE code which models the process. It must act as a training simulator for the scientific methodology, and it should provide scientific visualization of output data resulting from either research model. Since the software layer must handle both models in a unified way, it is clear that they must be tied together from a theoretical viewpoint and from an operational viewpoint. The input and output associated with both models must appear in a uniform way to the scientist using the virtual lab.

It is not clear, a priori, that the two research models are coupled in this manner. Generally, the computational code that models the experiment is a fortran program driven by a fortran input file, where the input data bears little resemblance to the experimental input and process that it is supposed to represent. In addition, when the experimental input or process is varied, it is often unclear how to modify the input data to reflect the changes. The uncoupled state of these models is a significant problem for the implementation of a SoftLab laboratory.

When the models are sufficiently coupled for a given scientific environment, the design of the virtual laboratory can begin. In order to support the various activities that take place in an physical laboratory, scientists must be able to use the virtual laboratory to

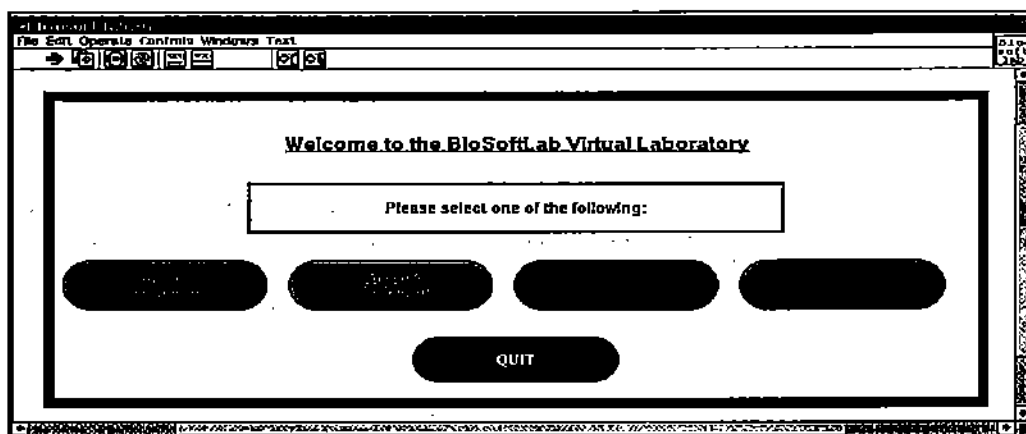
- perform physical experiments by interacting with and controlling remote instruments,

- simulate physical experiments through computer modeling of the experimental process,
- compare physical and simulated data by playing back recorded experiments, and
- train scientists to understand the physical experiment, the computational model of the experiment, and the relationships between them.

A choice of these four *scenarios* is available to scientists after they have selected their virtual laboratory at the SoftLab gateway.

---

FIGURE 2. Four scenarios of the virtual laboratory



The graphical interface which is then presented to the user is a software representation of the physical laboratory. Each important physical device is present, in particular, all instruments and equipment used during the experimental process must have a visual representation in the virtual laboratory. These software devices will henceforth be called virtual instruments.

Each scenario has a specific functionality that relates to a basic activity occurring in the lab. In the *physical experimentation scenario*, scientists can set up and control the laboratory instruments and experimental process from a remote location, and they can monitor the experiment via animation of the virtual instruments. The animation may be driven by state reports from the physical devices, or by calculations based on the input data. They can control data collection and extraction remotely during the experiment, and they can visualize the experimental results at its conclusion. The virtual lab can also act as an "experiment database", where the experiment's input configuration and output results can be saved for later analysis. The option of naming and saving experiments is truly a benefit for the scientists, since information defining the experiments will now automatically be available on-line to browse through, catalog, and analyze.

In the *virtual experimentation scenario*, scientists will set up the virtual instruments and experimental process just as in the physical scenario. Afterwards, the physical setup is transformed to the input required for the computational model. Additionally, parameters that are strictly numerical will be specified via special interfaces, and an expert sys-

tem will be on hand to query for process characteristics or computational parameters. During the processing of the computational code, the virtual instruments will receive intermediate results so that they can be animated to show the progression of the simulation process. Results can be visualized exactly as in the physical case. The simulated experiment may also be saved to the experiment database. Here, too, the input configuration and output results will be used to define the experiment in the database. Since the physical and simulated experiments now reside together, computations on their associated data can easily be done for comparison or analysis.

The *playback scenario* uses the experiment database, from which physical and simulated experiments can be retrieved. Scientists can review or compare experiment input configurations and selectively display output results. Physical and simulated experiments can be compared or combined, and multiple data sets can be visualized concurrently. Physical results can be used to improve the computational model via feedback processes, and simulation results can be used to design new physical experiments. Finally, scientists can “playback” an experiment on the virtual instruments. The functionality of the playback scenario links the two research models in new and important ways, and thus fulfills one of the main objectives of the SoftLab project.

The *training scenario* has two distinctly different audiences. First, it can be used by novice scientists to learn about the physical layout of the laboratory, the instrument characteristics and operation, and the experimental methodology. Novice scientists can also learn about the computational model associated with the experimental process, and how the experimental process and its simulation interact. Second, it can be used by more experienced researchers to learn how Softlab can be used as a virtual laboratory. The eventual use of SoftLab’s virtual laboratories in engineering classrooms is a major goal of the Softlab project.

---

### 3.0 The Software Infrastructure of SoftLab

---

An essential core function of the SoftLab laboratory is the representation and operation of the laboratory instruments in the virtual environment. Instruments look like and operate just as their physical counterparts. Each scenario presents this virtual laboratory with working virtual instruments.

These virtual instruments can be programmed for running a physical experiment, for running a simulation of the physical process, for training in the operation of the instrument, or for playing back an experiment. For simulations, the virtual laboratory finds and executes the code that represents the appropriate PDE model of the process. During a physical experiment, the virtual laboratory will communicate with and control the remote physical instruments. In all cases, the instruments are animated to show the progression of the process, physical or simulated.

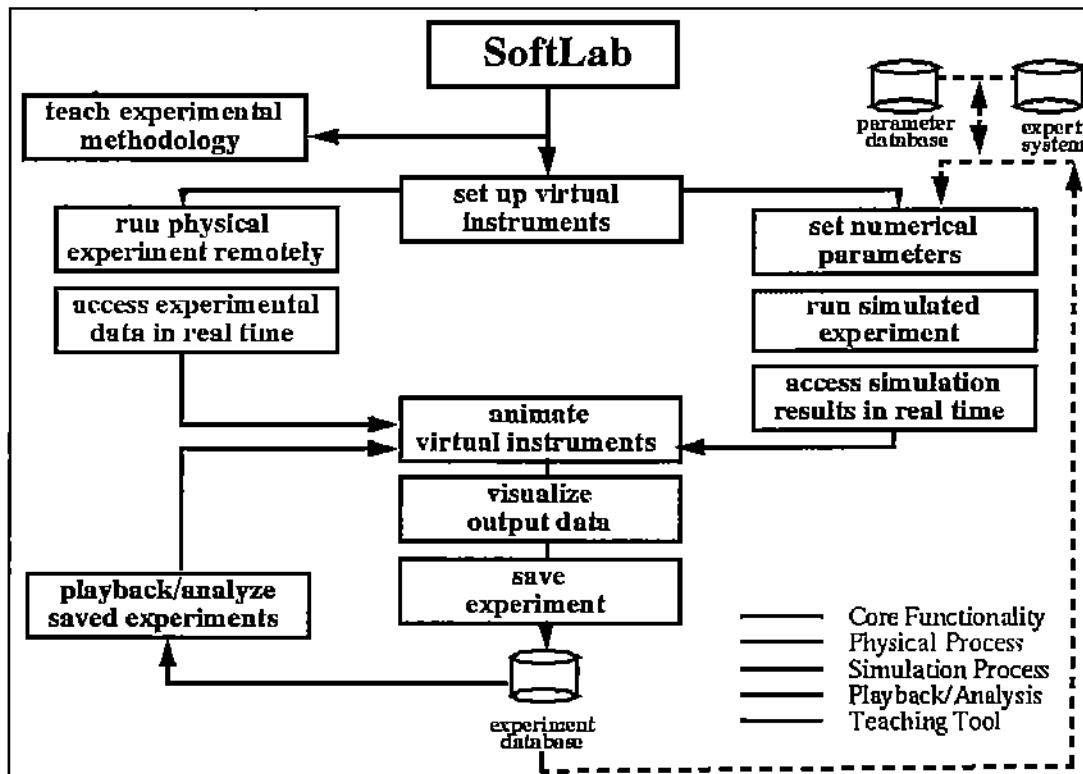
Access to an experiment database for storage, retrieval, and analysis of experiments is another core operation in the virtual lab. Since these functions are required throughout all four scenarios, a common database interface is provided.



Some functionality belongs to a restricted subset of the scenarios, such as visualizing results in real time. Other functionality is specific to a single scenario, such as the multi-media annotations used in the training scenario.

The specialized components are implemented as building blocks with specific input and output, and the components can be wired together and interact with each other according to the requirements of the scenario. The operation of each scenario is thus determined by the functional components that have been wired together. This also guarantees that the look-and-feel of the environment is uniform across all scenarios. A diagram of the functional components that support the four SoftLab scenarios is shown in Figure 3.

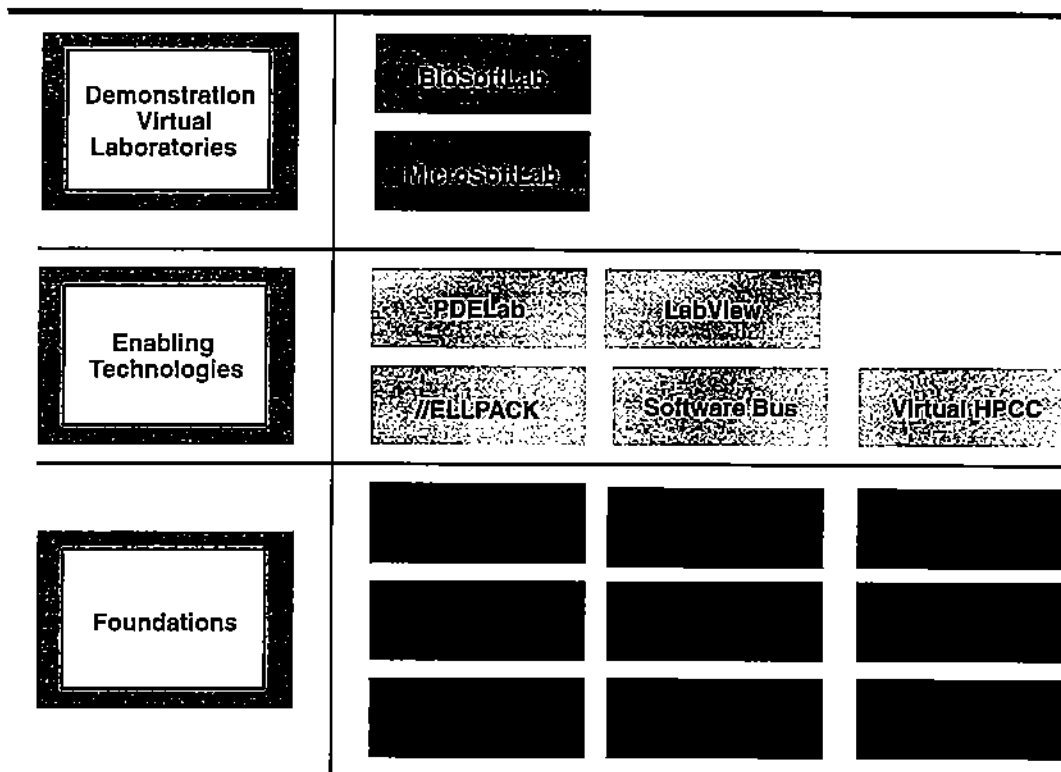
FIGURE 3. The functional components of SoftLab : integrating physical experiments and numerical simulation in a single, unified environment



To construct the building blocks for a virtual laboratory with the functionality described above, a kernel of software parts was required. The current realization of this kernel is shown in Figure 4 as a multi-layered software architecture.

At the base we have the low-level building blocks which support both the physical laboratory view and the PDE view (computational model) of the experimental process.

FIGURE 4. Components of the kernel used to build a SoftLab virtual laboratory



In order to formulate PDE problems, the necessary tools and facilities include geometric modeling, symbolic processing and intelligent advisors. Solving PDE problems requires libraries of generalized and domain specific solvers. For the laboratory processes considered for SoftLab implementation, the PDE models cannot usually be solved using general methods. They rely instead on customized code. Since the performance of such code is often measured in days, it is essential to introduce high performance (parallel) computing methods to speed the computations. Graphics display systems are needed for visualization and animation of the solutions and performance data.

To support the remote control of physical instruments, a specialized visual programming language is needed to build the visual representation of the laboratory instruments and to introduce functionality into the "knobs and levers". In addition, the instrument-specific protocols needed for communication and control must be accessible.

Other low-level functionality is required for a complete implementation of the four scenarios: a database management system to save and retrieve experiments, multimedia

technology to support audio/visual training and audio/visual communication with the remote laboratory, etc.

The algorithm, language, and system infrastructure demands for building such an environment are great, but the existence of the enabling technologies listed in the next level allowed the SoftLab virtual laboratories to be designed and implemented at a meta-level. Most of the low-level functionality was present in one or more of the problem solving environments (PSEs) in the second level.

PDELab[5] and //ELLPACK[8] provided the PDE problem solving environments for specifying and solving the PDE process model, and then visualizing the solutions. These PSEs have been designed and developed to enable users to solve PDE problems. For example, most domain specific (customized) solvers can be integrated into //ELLPACK quite easily. This allows users of the customized code to take advantage of the symbolic processing, geometry builders, mesh generators, and visualizers that are available within the PDE PSEs. Much effort has also been devoted to techniques for parallelizing sequential code. These tools and facilities already exist for the solving the PDE models that represent the experimental process.

The LabView[2] environment was used to build the visual representation of the virtual instruments and give them "working parts". The LabView visual programming language was found to be robust and complete. It was used to build all the necessary working instruments for the two demonstration laboratories, BioSoftLab and MicroSoftLab. Communication protocols for some instruments can be obtained through LabView. For those that aren't (our were not), the conversions from the virtual instrument user input to the physical instrument input must be done manually. This can only be accomplished when the instrument protocol is released by the manufacturer of the instrument.

When various tools and facilities within the PDE PSES needed to share information, or when the virtual laboratory needed to communicate with the PDE PDEs (or visa versa), the Software Bus[9] was used. The Software Bus provided the mechanism wherein tools (or facilities) inter-operate together without having explicit knowledge about each other. It also provided the infrastructure necessary for managing a set of distributed tools. For our purposes, the Software Bus was already a functional component of PDELab, but we needed to apply the Software Bus concept to the sharing of information between the environment governing input/output for the virtual instruments and the environment handling the PDE process model associated with that I/O.

Since no existing tool covered the multimedia requirements of our scenarios, a new multimedia component was built to support SoftLab. This component is now available as an "existing technology" (multimedia PSE) for future virtual laboratories and even more general applications.

The remainder of this paper is as follows: a detailed description of the implementation of the virtual environment for the Bioseparation Laboratory, a brief description of the implementation of the Micro-electronics Laboratory, a summary of the functionality of SoftMedia - the multimedia component of SoftLab, and the conclusion.

## 4.0 BioSoftLab

---

### 4.1 The physical laboratory and experimental process

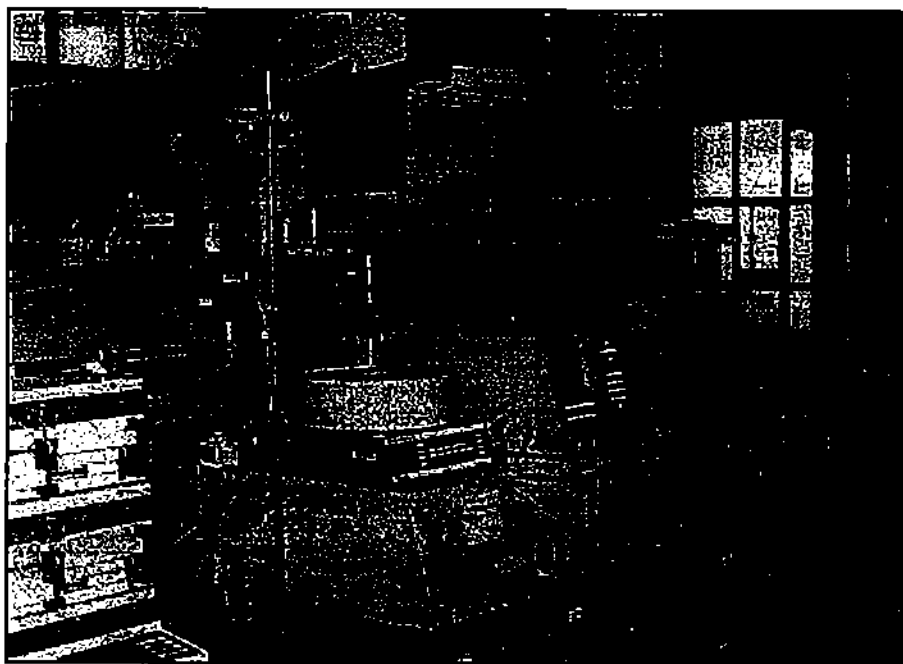
The Bioseparation laboratory is located in Purdue's School of Chemical Engineering and is operated by Professor Linda Wang. Our understanding of bioseparation research was a result of many discussions with Zidu Ma, a visiting faculty member. The physical laboratory is shown below.

Bioseparation is the process of separating chemical components by passing solution mixtures through an adsorbent column. The column is packed with sorbents selected for their adsorption properties with respect to the incoming solutions. Solutions pass through the column and elude from the bottom. Since each solution component adsorbs to the surface of the sorbents in a unique way, components elude at different times. Reactions within the column may also produce new components. The primary goal of bioseparation is to control what happens in the bioseparation column, so that one can predict when and at what concentration certain components will elude from the column. This process is used for final purification of proteins, chemicals, and biochemicals used in the manufacture of pharmaceutical and food products, for water treatment, and for many other biochemical processes. [3]

---

FIGURE 5.

The Bioseparation laboratory



The physical instruments required to support this research are from the Waters Chromatography Division of Millipore. The bioseparation column is the focal point of the

experimental process. Solution mixtures enter the column from two sources. One source is an injection syringe mounted on a tube inserted in the column. The other is a Waters 600E pump controller which regulates flow into the column from a combination of four attached reservoirs. When solutions elude from the column, component concentrations are measured by the Waters photodiode array detector. The measurements are collected by an attached NEC 80286 computer which Waters provides with custom hardware and software for communicating with the detector.

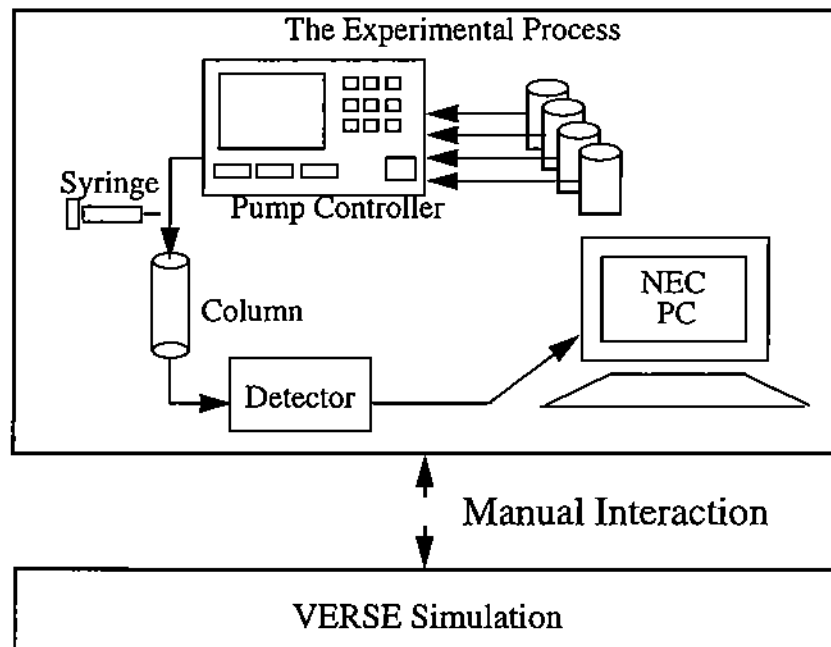
Because bioseparation experiments are both expensive and complex, computer simulation of the process is an attractive alternative to performing experiments. VERSE is a custom implementation of a numerical simulator which models the process, and this code is used by bioseparation scientists to simulate experiments. [4]

#### 4.2 A virtual laboratory for research, education and practice

BioSoftlab was designed with one goal in mind: merging of the two environments, "wet" and "dry", into one common interface. Before BioSoftLab, the experimental process and the corresponding simulation were separate and independent activities; there was no substantive link between these two scenarios. Input to the simulation was completely unrelated to the setup of the experiment which it represented. No automated method existed for comparing experimental output to simulation output, and any procedures for transferring or feeding back information from one scenario to the other were manual.

FIGURE 6.

Diagram of the experimental process and its relationship with the simulation



For us to better understand and implement this common interface, we had to acquire a plethora of information relating the experimental process and the simulation process. By grouping actions together and relating them to the virtual instruments, the bioseparation scientist can now see how different parameters of the simulation input relate to the physical input. This correlation, which will become obvious in later sections, was not obvious prior to the development of BioSoftlab.

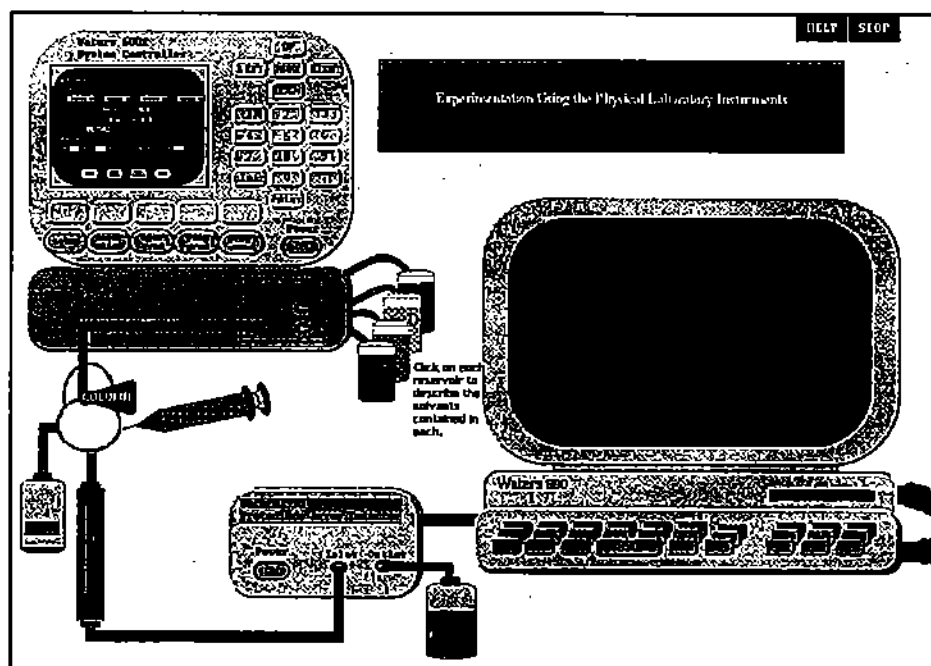
During the implementation of this PSE, it became obvious that the new environment could offer more than a common interface to the two environments. BioSoftlab could offer a means of communication between the two: data could be passed back and forth and information could be stored for later access and comparison. To support this functionality, the playback and analysis scenario was added.

#### 4.2.1 Using BioSoftLab to perform experiments

In order to run a physical experiment using the BioSoftlab interface to the physical lab, select the *Physical Experiment* scenario from the BioSoftlab main menu. BioSoftlab presents you with a software view of the physical lab and its equipment. The functionality of each virtual instrument is identical to that of its physical counterpart.

FIGURE 7.

Entering the physical experimentation scenario



In order to setup and run a physical experiment, the user has to set up the physical lab; this includes filling up the reservoirs and the injection with the solutions used in the experiment. Now scientists may perform the rest of the set up remotely using the Bio-

Softlab interface to the physical laboratory. There are various types of information required by BioSoftlab:

1. **Enabling the equipment.** Enabling the detector, enabling the controller, deciding whether to clear the loop or to perform an experiment.
2. **Programming the equipment.** Programming the (virtual) controller in the same way one would program the physical Waters 600E Controller in the lab; selecting the operating mode of the (virtual) controller, specifying the flow rate of each solvent for each time step, etc.
3. **Setting up the PC for data acquisition.** Running the data acquisition software on the (virtual) PC, specifying how the data should be gathered during the experiment.

This information is identical to the information the user would input to the actual equipment in the physical lab in order to set up a physical experiment. Since this input is sufficient to setup and run a physical experiment in the lab, it should also be sufficient to run a physical experiment using BioSoftlab. This is indeed the case.

Given this input, BioSoftlab has all the information required to run a physical experiment. After the experiment is started, BioSoftlab displays an animation of what is happening in the physical lab: the reservoirs empty, liquid flows through the column, etc. Currently, this animation information is calculated given the controller flow rate information. As the next step, we would like to communicate with the controller to acquire this information and display it in the same way on the BioSoftlab virtual instruments.

When the experiment is complete, BioSoftlab can be used to connect to the Waters PC and access the data which was collected. This data is displayed on BioSoftLab's virtual PC. After reviewing the data, the scientist can choose to record the experiment in an experiment database for later retrieval. The information saved in the database includes both the experimental input and output. For the information to be complete (so that we can re-create the experiment or compare it to some other experiment) we ask the user to provide the following information:

1. **Description of the physical makeup of the input solutions.** In order to save the experiment in the database, information like the quantity of each solution used and a description of the chemical composition of each solution is required to fully describe the experiment.
2. **Physical description of the equipment used.** Equipment like the tube connecting the column to the controller and the detector are non-standard, and different tubes could result in different data being generated. We request that the user provide all information relative to the tube, so that re-creating the exact experiment is possible.

The above data is information that an experimental scientist would need to record (in a notebook) to identify or re-create an experiment in the physical lab. Therefore, the information is also required here to fully identify an experiment within the database. Thus, BioSoftLab's database also becomes an "electronic notebook" or experiment depository.

During the implementation of this scenario, we encountered various problems related to the passing of information between the controller and BioSoftlab. This was particularly hard, since the Waters equipment was not designed to allow for remote control. There-

fore, control via the virtual instruments was possible only by translating the user's input to the actual controller language. The ease of implementing a physical scenario, therefore, is directly related to the design and functionality of the actual physical equipment.

Another problem was more specific to interaction with the user. The virtual environment must be able to request the necessary information in such a way that even novice bio-separation scientists could supply the required input. The interaction between physical equipment is automatic, and we had to simulate the same inter-connection. If the user inputs a certain piece of information then we should be able to derive from it as much as possible, without asking the user to supply more than is absolutely necessary.

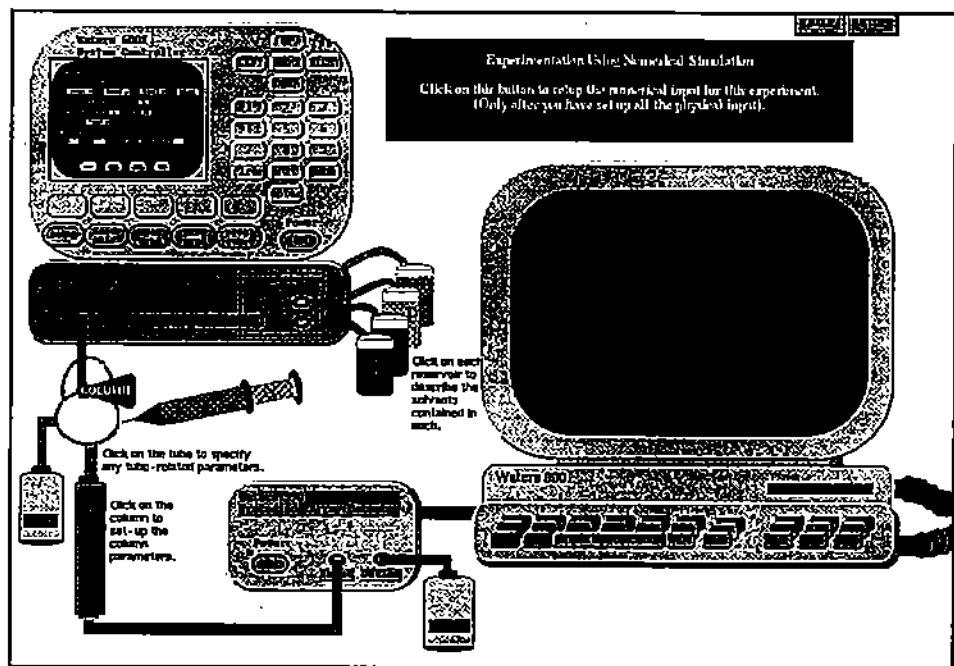
#### 4.2.2 Using BioSoftLab to simulate experiments

In this scenario, the experimental process and the computational model which represents it are linked together and interact with each other in a truly unified manner.

To simulate an experiment numerically using the VERSE simulation code, select BioSoftLab's virtual mode. BioSoftLab presents you with a software view of the physical laboratory and its equipment. The instruments will be set up and programmed exactly as if for a physical experiment; the functionality of each virtual instrument is identical to that of its physical counterpart. However, in the virtual mode, the physical input is then transformed to the input required by the computational model.

FIGURE 8.

Entering the virtual experimentation scenario





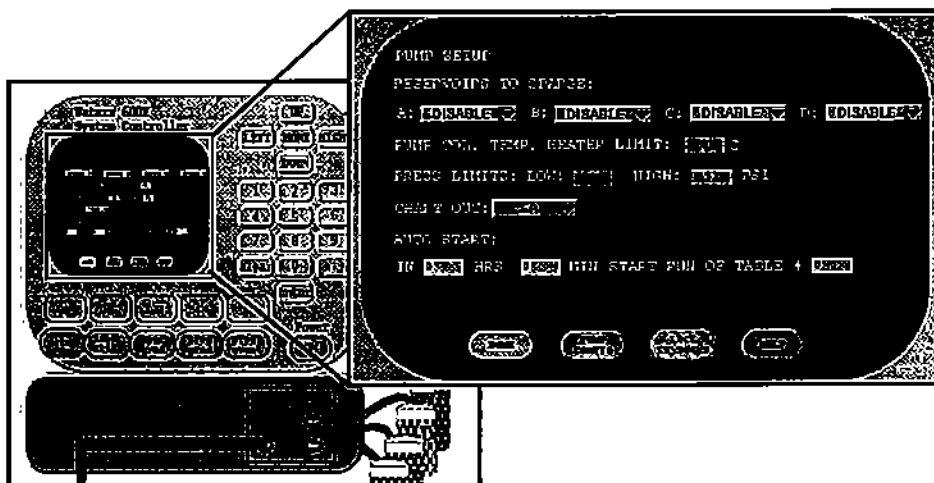
Since there are additional numerical parameters and system characteristics associated with simulation, special interfaces built into BioSoftLab's virtual mode allow you to control the additional specifications. This information consists of the following:

- specification of the physical makeup and characteristics of input solutions and sorbents,
- properties of the input solutions needed by the computational model,
- identification of the reactions (if any) amongst the solutions, that will occur in the column, and
- selection of the process model and associated numerical parameters.

With the physical and numerical input complete, the VERSE simulator is started. Simulation output is collected in real time, and intermediate and final results generated by the simulation can be visually represented on BioSoftLab's virtual instruments. Data can be saved in various formats, and can be loaded into PDELab for three-dimensional scientific visualization and animation. When the simulation has finished, BioSoftLab allows you to "save" the experiment to an experiment database. Saving a virtual experiment means that the input configuration and output results are stored so that they may later be retrieved to playback the experiment or to view intermediate and final results.

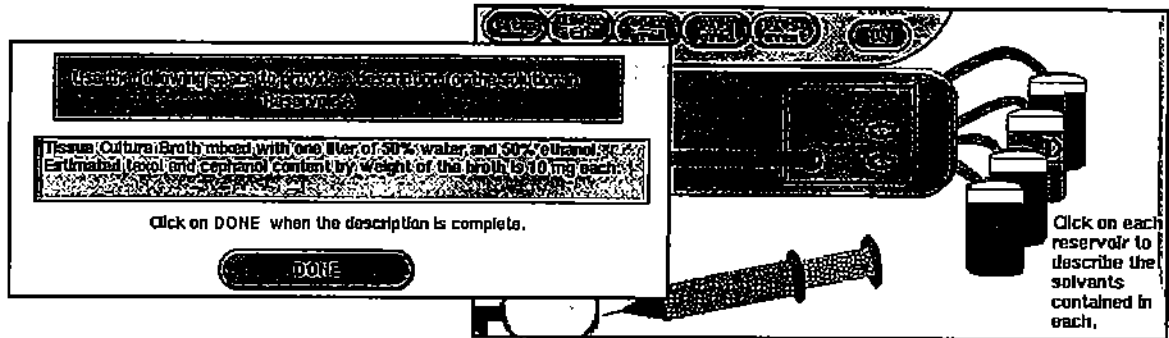
FIGURE 9.

Instrument setup: programming the virtual pump controller



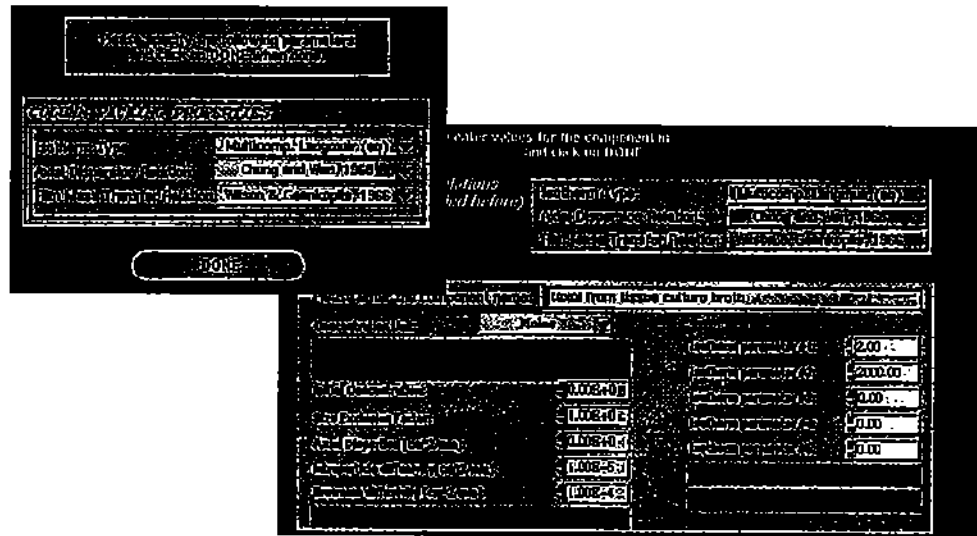
In order to implement the above scenario, we encountered some major difficulties. We had already built the software representation of the physical instruments, and these virtual instruments were fully "functional". However, there existed no mapping between input/setup for the physical experiment and input to the computational model which was used to simulate it. We had to identify and separate the kinds of information required by the simulation code. A clear understanding of the differences between the following categories of data enabled us to devise a stable and consistent mapping.

FIGURE 10. Solution input: filling reservoir A with tissue culture broth



1. information related to the experimental procedure and instrument setup. This information should be entered exactly as it is in the physical experimentation scenario.

FIGURE 11. Numerical properties of a component: describing taxol numerically



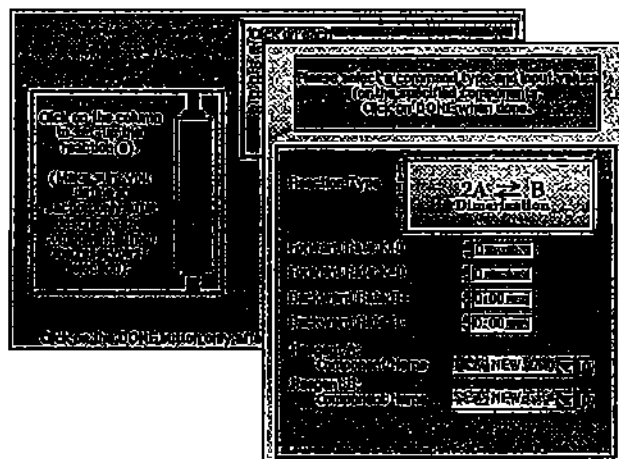
2. information related to the physical makeup and characteristics of the input solutions and sorbents. Here, the simulation setup must depart from the physical setup. In the physical case, the experiment is driven by the characteristics, properties and makeup of the solutions and sorbents themselves. In the simulated experiment, the physical makeup and properties must be described numerically to the simulator. Therefore, users must specify the contents of the solutions and their physical characteristics. If a reservoir is filled with a “taxol broth” solution, the user must name (or tag) the solution and describe exactly what is contained therein. Clearly, users should not be required to identify the contents of tagged input more than once.

Therefore, this information required the introduction of a database.

3. **information related to the properties of the input solutions which are used to generate numerical parameters.** There are several different examples of this type of data. The most important is the specification of which components from the input solutions are to be simulated. These are special components whose journey through the column is to be monitored by the simulation process. Special information about these components is needed (for example, isotherm coefficients, diffusivity, etc.). This again requires a database interface for storing and retrieving such data.
4. **reaction information.** Reactions occurring in the column require particular attention. These are considered "components to be monitored", and the component information described above is needed here as well.

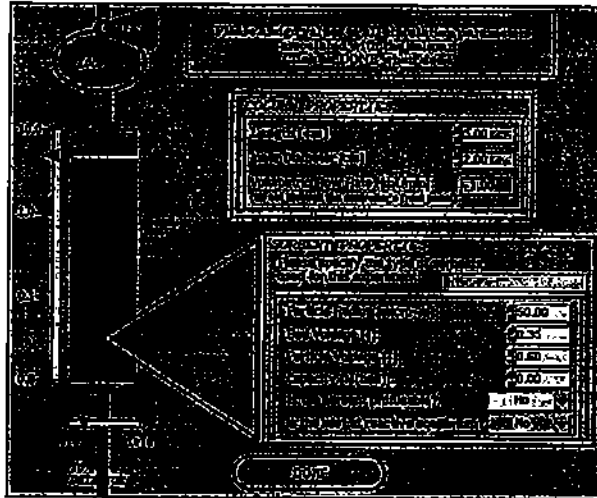
FIGURE 12.

Reaction specification: identifying a dimerization reaction



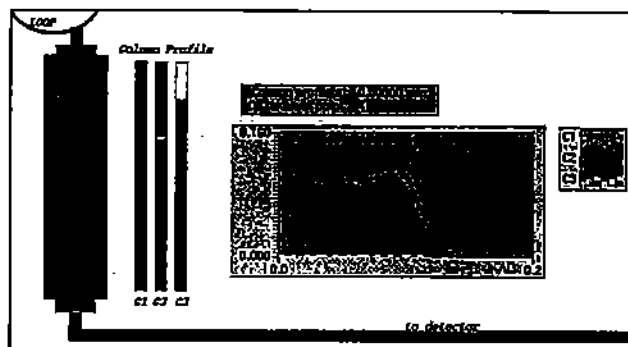
5. **process model specification with associated numerical parameters.** This refers to the system of partial differential equations that will be solved by the simulator, along with the numerical parameters required by the Finite Element Method (FEM) which is used to solve them. There are several "experiments" or "computational models" within VERSE, and the application of a given model is governed by the type of experiment which is being simulated. Ideally, expert system advice for the proper process model should be available. Other FEM parameters, such as the mesh specification, should be tied very closely to the column - since the simulation actually represents what occurs in the column during the experiment. To emphasize the relationship between the physical and simulated experiments, the numerical parameters are tied wherever possible to the virtual bioseparation column.
6. **specification of monitoring and output requirements.** The simulation produces concentration measurements of the specified components over time. This data can be generated for concentrations at the end of the column or at specified points along the length and radius of the column.

FIGURE 13. Packing the column



Probing the depths of the relationships between the physical process and the computational model required many refinements to our virtual environment. Since the theoretical model allows far greater flexibility in the specification of input than the experiment itself, we had to extend the concept of physical input to allow the experimenter to scale-up, scale-down, optimize, mix or separate the input solutions in ways that were not feasible in the physical laboratory. Since, however, these parameters represented extrapolations of actual experimental input, they could easily be tied to the physical devices from which they were extrapolated. In this way, users of the numerical simulation model could understand how each parameter of the theoretical input was associated with the actual experimental process.

FIGURE 14. Monitoring the column during simulation

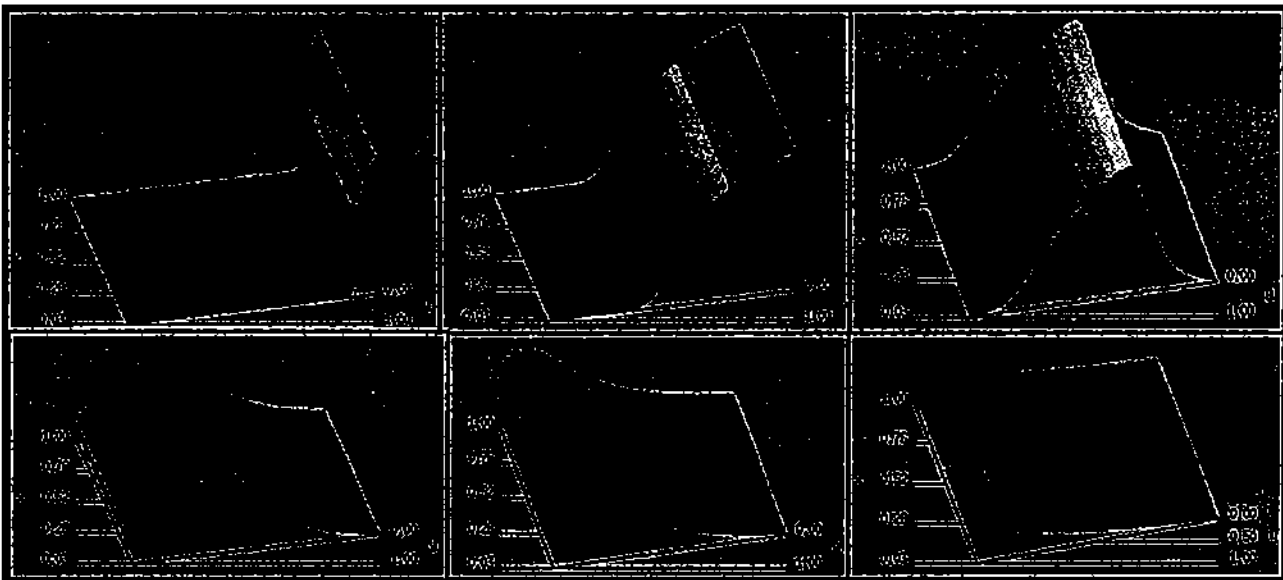


There were also differences in the way physical and virtual experiments were visualized in BioSoftLab. The simulated experiment is monitored and visualized directly on the BioLabView instruments, just as in the physical experiment. But in this case, the data

used to “animate” the instruments is generated by the simulation instead of by the actual devices.

This again allows the experimenter greater flexibility in visualizing the experiment, since the simulation generates intermediate data which the physical devices cannot produce. Column profiles generated by VERSE produced three-dimensional time-dependent component concentration data along the length of the column, which we visualized on special column-graphs within BioSoftLab. We also animated this data using scientific visualization tools in PDELab. [5] Graphs such as these were not possible before the advent of BioSoftLab.

**FIGURE 15.** Three-dimensional view of a reactant's concentration as it moves down the column. In this case, the concentration is uniform across the column width.



#### 4.2.3 Using BioSoftLab to playback experiments

The playback tool was designed to enable scientists to re-run experiments for which the input configuration and the output data already exist. BioSoftLab users could already run both the physical and the virtual scenarios for the same experiment within the same environment. It seemed an obvious extension to provide them with the ability to re-run experiments, either simply to review the input and generated output, or even to compare the input and output of the physical experiment with that of a virtual run.

In order to be able to re-run experiments, we had to construct a database where bioseparation experiments could be stored, and we had to define a format for the stored data. As a result, the user can now choose to save an experiment from either the physical or virtual modes. After the user specifies the input to the experiment, BioSoftLab collects all the information necessary to identify the experiment and records it in a file with pre-defined format. This includes physical characteristics of the solutions, physical dimen-

sions of the column and the tube, etc. This data file is stored in the database where it can be retrieved at a later time. If the user is performing a virtual experiment, the data file generated by BioSoftLab (and used as input to the simulation module, VERSE) will also be stored in the database.

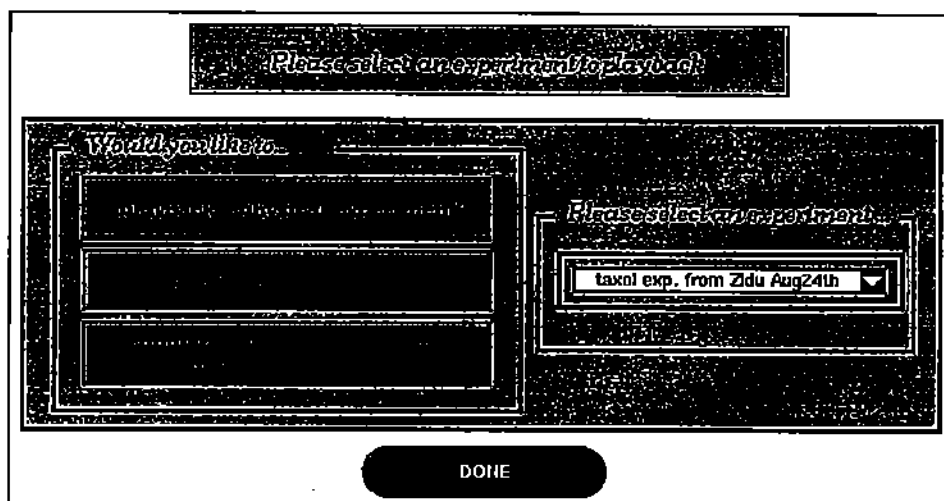
Running the physical experiment or the simulated experiment generates various mode-dependent data files. In the case of a physical experiment, if the user opted to retrieve the detector-generated data from the PC, a file showing the concentration at the end of the column is the only file output during the experiment. In the case of a simulated experiment, the user specifies whether a history file (time-dependent end of column concentration) and/or a profile file (time-dependent concentration along the entire length of the column) should be generated by VERSE, as well as whether PDELab format files should be generated. These output files are all stored into the database together with the input file.

With all the files required to define an experiment now stored within the database, the user is able to re-play an experiment. Selecting the Playback mode from the BioSoftLab menu, presents users with the following choices:

- **Playback a physical experiment.** This allows users to select an experiment from the database for which data from a physical experiment exists.
- **Playback a simulation.** This allows users to select an experiment from the database for which simulation data exists.
- **Compare physical and simulated data.** This allows users to select an experiment from the database for which both physical and simulation data exist.

FIGURE 16.

The playback choices

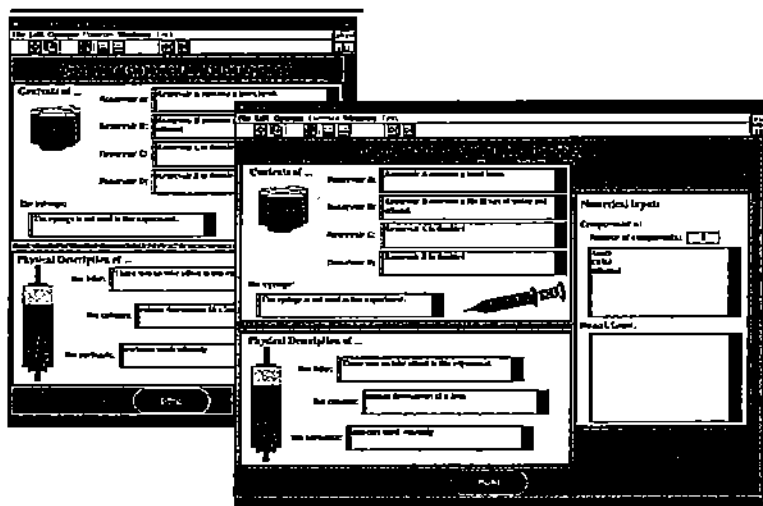


After the user selects an experiment for playback, BioSoftLab displays the input to the experiment: physical or virtual. In the case where users are comparing physical data to simulated data, both the physical and simulation input will be displayed. The virtual

input is displayed in a format similar to that of the physical input, except that information about the components and reactions is also included.

If the user is simply reviewing the output data corresponding to a virtual scenario, the column profile data will be displayed (if it has previously been saved). Next, in both the physical and the virtual scenarios, an animated histogram of the concentration at the end of the column will be displayed on the virtual PC monitor. In the hybrid case, when the user is comparing physical data to the simulation results, both the detector-generated histogram and the corresponding VERSE-generated histogram will be displayed on the virtual PC, so that the results can be compared.

FIGURE 17. Comparison of the input to the physical and virtual scenarios



Even though the actual input and output is displayed as part of the Playback tool, there is currently no information saved which shows how an experiment progresses over time. A next step would be to enhance the database so that additional intermediate results could be stored which would allow bioseparation scientists to “run” an actual animation of the experiment, instead of viewing a sequence of static presentations of the data.

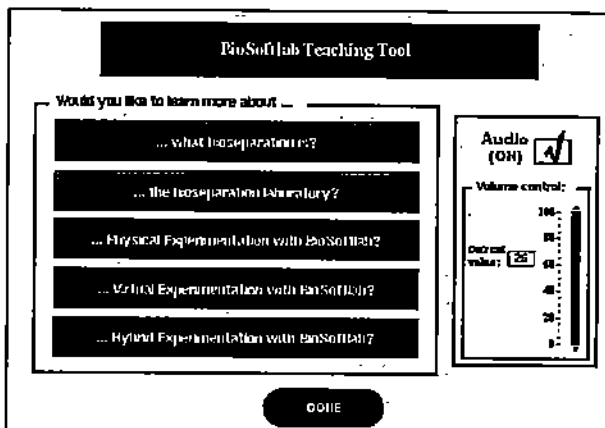
#### 4.2.4 Using BioSoftLab for bioseparation training

BioSoftLab’s Training Tool targets a wide spectrum of users, from novice bioseparation students to bioseparation research scientists. The Training Tool can be used to

- learn about the lab setup and instrument control,
- understand the actual bioseparation process,

- find out how BioSoftLab can be used to setup and “run” a physical bioseparation experiment, and
- learn how to use BioSoftLab to run the computational model.

FIGURE 18. Teaching Tool selection menu



If the Teaching Tool is selected, users are presented with the following options:

For the bioseparation novice:

- **What is bioseparation?** Various multimedia annotations are used to give the user an understanding of what a bioseparation experiment involves and what it may be used for.
- **The bioseparation laboratory.** A description of the instruments used to perform a bioseparation experiment is presented.

For the bioseparation scientist:

- **Using BioSoftLab to perform a physical experiment**
- **Using BioSoftLab to perform a virtual experiment**
- **Using BioSoftLab to perform a hybrid experiment**

When the bioseparation scientist selects one of the three choices listed above, he or she is guided through a tour of setting up and running an experiment using BioSoftLab. Using a wide range of multi-media annotations, the user is taught the step-by-step routine required to set up an experiment using BioSoftLab.

If the user is already familiar with the actual bioseparation experiment, he or she can use BioSoftLab's Teaching Tool to learn or practice setting up a physical or virtual experiment and better understand the interaction between the physical and computational models.



Clearly BioSoftlab can be used in an engineering classroom as an instructional aid for teaching

- the physical layout of the laboratory,
- the equipment characteristics, operation and functionality,
- safety in the laboratory,
- experimental methodology, including how to monitor the devices and collect results,
- use of the computational model to simulate an experiment, and
- the interaction between the experimental process and its simulation.

All of the above can be done from within the classroom, without entering the laboratory where the experiment actually takes place.

The advantages of having this capability in the classroom are obvious. It increases expertise in laboratory instrument operation on a large scale and without use of the costly lab. It allows access to laboratories that may not be locally available. It allows beginners to perform "dangerous" experiments. It uses a combination of recorded actual and simulation data to understand how these two models interact. It uses simulation to visualize and explain phenomena not measurable by real-time experimentation. These and many other benefits can result from a successful implementation of BioSoftlab.

---

## 5.0 MicroSoftlab

---

### 5.1 The physical laboratory and experimental process

The Microelectronics laboratory is located in Purdue's School of Chemical Engineering and is operated by Professor Christos Takoudis. Our understanding of chemical vapor deposition research was a result of discussions with Christopher Panczyk, a graduate student in Chemical Engineering.

The experiment involves the chemical vapor deposition (CVD) of silicon from silane on Si(100) substrate surfaces (wafers). *In situ* infrared spectra emissions are collected real-time to characterize the silicon surface and study reaction intermediates during the growth of a thin film of Si(100) at 750 degrees C and 10.0 Torr.[6]

The microelectronics laboratory consists of a reactor chamber (cell) that is placed inside a Nicolet 800 FTIR spectrometer. The cell was purchased by SpectraTech and modified to allow for better temperature control vacuum capabilities. Auxiliary equipment needed to supply the emission cell with N<sub>2</sub>, SiH<sub>4</sub> and cooling water along with all temperature, pressure and mass flow instruments remain on the outside of the spectrometer.[7]

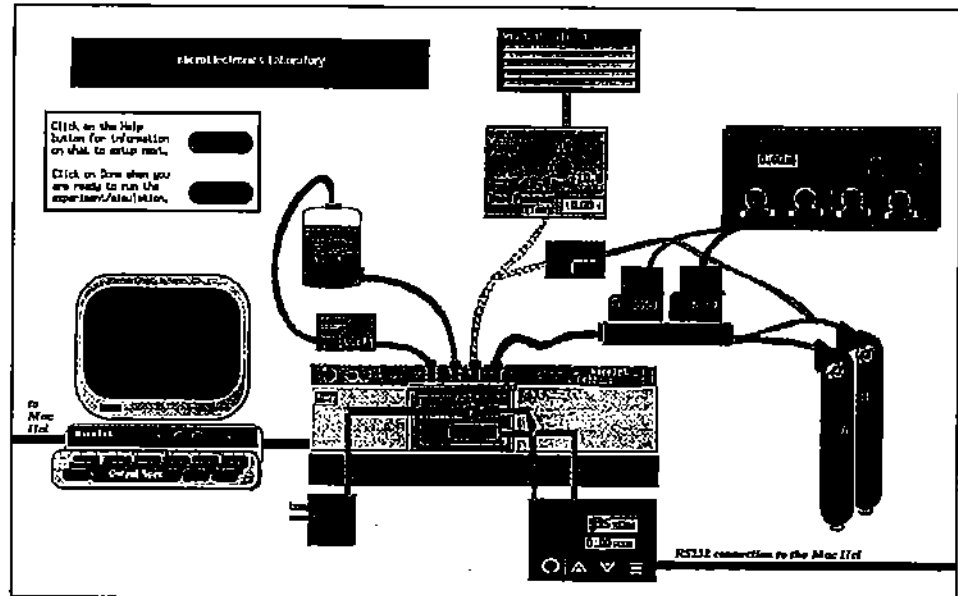
### 5.2 A virtual laboratory

The MicroSoftlab laboratory is designed in the same way as its BioSoftlab equivalent. A virtual instrument corresponds to each physical instrument, and the user can operate

these instruments to remotely control the experiment. Since the actual experiment required specifying information for the FTIR, through the Nicolet, every 10 to 15 minutes, MicroSoftlab provides the microelectronics scientist with the ability to preprogram most of this information. MicroSoftlab would then calculate the required data from the

FIGURE 19.

The MicroSoftlab environment



specified input, and then supply the Nicolet with the necessary information every 10 to 15 minutes. This allows the microelectronics scientist to perform a physical experiment without being present in the lab throughout the entire process.

Only part of the design for MicroSoftLab has been implemented. Remaining tasks include the integration of the computational model and animation of the virtual instruments.

## 6.0 SoftMedia

An essential component of SoftLab is the availability of multimedia functions. The physical and virtual laboratories are separated from each other (perhaps by many miles), and scientists using the virtual laboratory environment may need to see what is happening in the physical lab. They may also need to communicate with scientists and technicians who are in the lab, and this communication may require video, audio or transmissions of diagrams and computations.

SoftMedia is a general framework for integrating multimedia conferencing tools to any problem solving environment requiring multimedia facilities. An instance of SoftMedia is a collaborative platform which supports audio conferencing, video conferencing, and whiteboard capabilities.

SoftMedia supports dynamic re-configurability, so that a single interface can be presented to the user across machines with varying low-level audio and video software. These off-the-shelf multi-media tools are used to configure SoftMedia properly, according to the communication requirements of the PSE.

SoftMedia has been integrated into Softlab, and provides a multi-media platform for collaboration between the physical and virtual laboratories. This allows users of Softlab to look at and monitor the physical laboratory via video conferencing while running a remote experiment using SoftLab's virtual laboratory. Scientists using the virtual environment can also communicate with experimental scientists in the actual lab using audio conferencing and whiteboard facilities.

---

## 7.0 Conclusions

---

The traditional prototyping approach involves the definition and construction of a physical system in terms of a descriptive set of specifications and properties followed by the construction of simplified numerical models for simulating the behaviors of that system under some set(s) of physical laws. This process often involves the use of diverse and separate techniques and tools. The user is often forced to manage multiple sets of data and computational environments, making the computational analysis "after-the-fact" rather than an integral part of the prototyping process. This approach is cumbersome, inefficient, restricted in scope, inflexible (since it does not allow the scientist or engineer to easily explore alternative formulations and designs), and impractical to use as training environment.

We have presented a framework called SoftLab that allows the seamless integration of physical (experimental) and simulated (computational) prototyping which is controlled by a single virtual environment with a user interface that can be adapted to the targeted physical process. We have demonstrated proof-of-concept in a chemical engineering laboratory associated with the bioseparation process. A SoftLab framework is currently under development for a microelectronics materials chemical engineering laboratory.

---

## 8.0 References

---

- [1] Christoph M. Hoffman, Elias N. Houstis, John R. Rice, Ann Christine Catlin, Margaret Gaitatzes, and Sanjiva Weerawarana, "SoftLab - A virtual laboratory for computational science", *Mathematics and Computers in Simulation* 36, 1994, pp. 479-491
- [2] National Instruments Corporation, LabView User Manual, Austin, Texas, 1992.
- [3] K. E. Van Cott, R. D. Whitley and N.-H. L. Wang, "Effects of temperature and flow rate on frontal and elution chromatography of aggregating systems", *Separat. Technol.*, 1991, pp. 142-152.

---

## References

---

- [4] J. A. Berninger, R. D. Whitley, X. Zhang, N.-H. L. Wang, "A Versatile Model for Simulation of Reaction and Nonequilibrium Dynamics in Multicomponent Fixed-Bed Adsorption Processes," *Computers in Chemical Engineering*, 15, 1991, pp. 749-768.
- [5] Sanjiva Weerawarana, Elias N. Houstis, John R. Rice, Ann Christine Catlin, Cheryl L. Crabill, Chi Ching Chui and Shahani Markus, PDELab: An Object-Oriented Framework for Building Problem Solving Environments for PDE Based Applications, *Proceedings of the 2nd Annual Object-Oriented Numerics Conference*, 1994, pp 79-92.
- [6] I-H. Oh and C.G. Takoudis, "Modeling of Epitaxial Silicon Growth from the  $\text{SiH}_2\text{Cl}_2\text{-H}_2\text{-HCl}$  System in an RF-Heated Pancake Reactor," *J. Appl. Phys.* 69, 1991, pp. 8336-8345.
- [7] I-H. Oh and C.G. Takoudis, "Modeling of Chemical Vapor Deposition in Pancake Reactor Systems," *J. Electrochem. Soc.*, 90-12, 1990, pp. 75-81.