1993

Softlab - A Virtual Laboratory for Computational Science

Christoph M. Hoffmann
Purdue University, cmh@cs.purdue.edu

Elias N. Houstis
Purdue University, enh@cs.purdue.edu

John R. Rice
Purdue University, jrr@cs.purdue.edu

Ann Christine Catlin

Margaret Gaitatzes

See next page for additional authors

Report Number:
93-061

SOFTLAB - A VIRTUAL LABORATORY FOR COMPUTATIONAL SCIENCE

Christoph M. Hoffmann
Elias N. Houstis
John R. Rice
Ann Christine Catlin
Margaret Gaitatzes
Sanjiva Weerawarana
N-H Linda Wang
Christos G. Takoudis
David G. Taylor

CSD-TR-93-061
September 1993
(Revised December 1993)

SoftLab - A Virtual Laboratory for Computational Science

Christoph M. Hoffman, Elias N. Houstis, John R. Rice, Ann Christine Catlin, Margaret Gaitatzes, and Sanjiva Weerawarana
Department of Computer Sciences

N-H. Linda Wang, Christos G. Takoudis, and David G. Taylor
School of Chemical Engineering

Purdue University
West Lafayette, IN 47907, USA.

Abstract

We describe a newly established research project called *SoftLab* in the area of computational science and computational engineering. The *SoftLab* project attempts to link physical laboratory experimentation with computer control and simulation to provide a virtual laboratory for computational science. We describe the overall project objectives and then introduce the three focus projects of *SoftLab*: Two Chemical Engineering SoftLabs (bioseparation and computational electronics) and a Mechanical Engineering SoftLab (computational mechanics). Preliminary results of our efforts are also described.

Keywords: Experiement Control and Data Collection, Partial Differential Equations, Numerical Simulation, Problem Solving Environments, Computational Geometry.

1. Introduction

In this paper we describe the goals of the newly established project *SoftLab* in the area of computational science and computational engineering, and report on early developments. The focus of SoftLab is to realize a virtual laboratory with a software layer above the familiar physical wet and dry laboratories of science and engineering. The project builds the needed software and algorithmic infrastructure to realize some of the important activities in a lab, and produces the electronic analog of a scientific laboratory environment for important applications. The real time requirements for the interactions between experimental or production processes on the one hand, and solving analytical models by computer on the other hand, necessitate addressing the fundamental challenge of harnessing the power of high-performance computing equipment and experimental instruments

and exploiting them through very high level systems that are problem-oriented. Such systems require a wide range of expertise plus a flexible and diverse array of equipment. The SoftLab project is to combine the hardware facilities with problem-solving environments (software facilities) to carry out scientific and engineering research. 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 computing power.
- A teaching laboratory providing exploratory courses that migrate cutting-edge research into the curriculum with access to state-of-the-art facilities. This laboratory will be the principal facility of a new interdisciplinary graduate degree program in Computational Science and Engineering.

The laboratory provides an ideal environment to accept and meet the challenges of computational science and engineering. The facility leverages the infrastructure and support bases we have built through the *Computing About Physical Objects* (CAPO) project. The project benefits from an environment with strong expertise in mathematical software, electronic prototyping, geometric modeling, parallel algorithms, databases, software engineering, and computer systems. The facility is a catalyst for significant research in computational science and engineering, both within the computer science department, as well as in other departments, and so accelerates the maturation of a number of collaborative, interdisciplinary projects that are beginning to create a critical mass of applications-oriented research and expertise in computational science. This effort is Softlab.

A central group effort within Softlab addresses the full range of issues in the future paradigm of computational science. It creates three *problem solving environments* (PSE): one each in computational mechanics, computational chemistry, and computational electronics. These involve a core group of computational scientists (Christoph Hoffmann, Elias Houstis, and John Rice) from the Computer Science Department and three scientists from the Schools of Engineering (Dave Anderson, Linda Wang, and Christos Takoudis).

2. SoftLab Research Agenda

The SoftLab research focuses on the creation of environments, tools and infrastructure needed in an electronic laboratory for computational science. A fully functional Softlab requires the creation of this software for two purposes: first to integrate diverse tools and infrastructure into application-specific problem solving environments (PSEs), and second, to harness the enormous computing power of well-coordinated parallel and distributed computations. There is overlap in the infrastructure for these two purposes.

Figure 1 shows a functional view of SoftLab as a multi-layered architecture. At the lowest level, we have the algorithms and systems infrastructure., the low level tools needed to build problem solving environments. Here, we are re-using the facilities developed within the CAPO project and also developing several tools that assist in building problem solving environments.

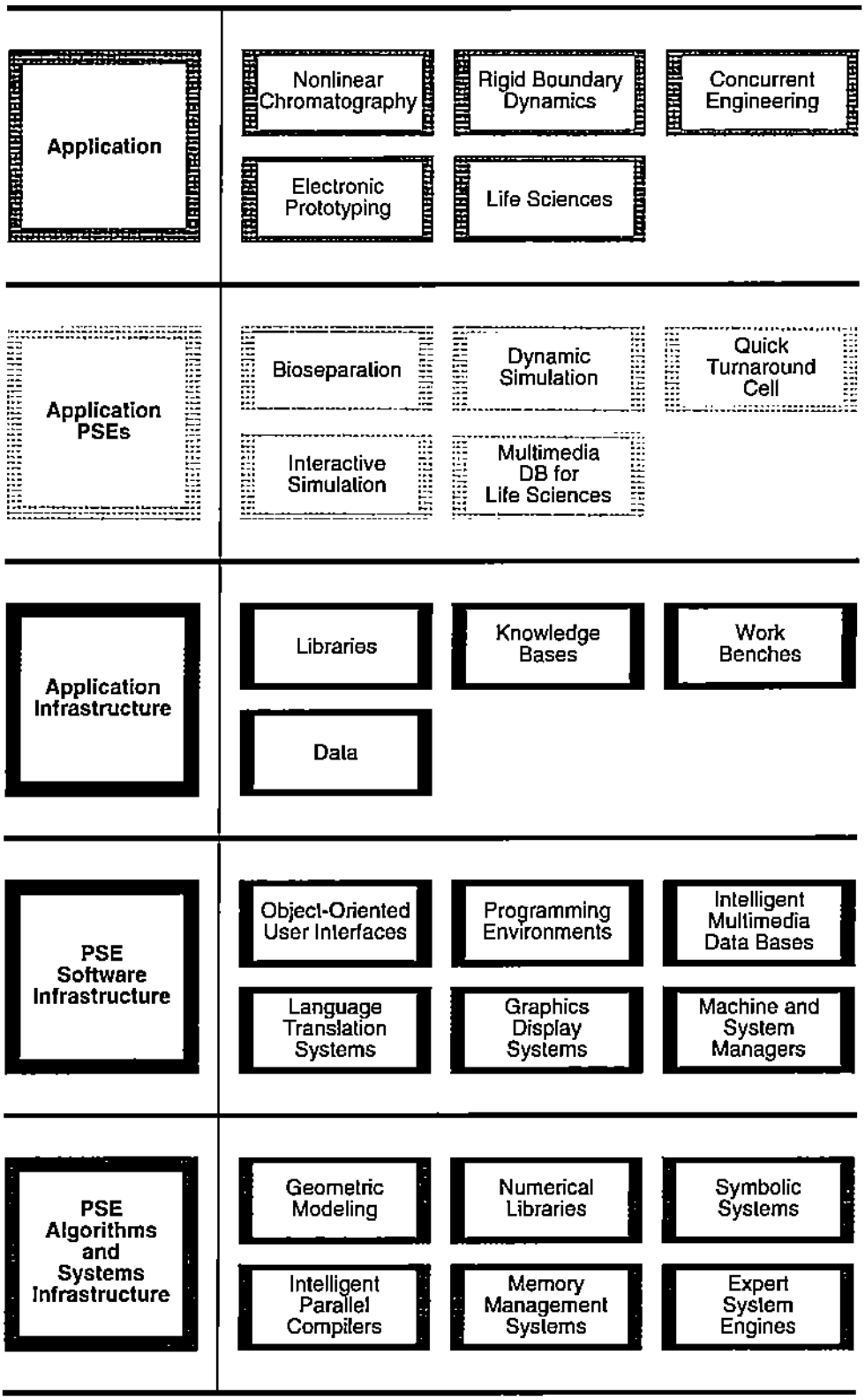

FIGURE 1. A functional view of SoftLab showing its five levels and components. The components at the top represent focus projects. The focus research also addresses building many of the infrastructure components shown at the other levels.

At the next level, we have the problem solving environment infrastructure. We have developed some software platforms here ourselves, as the available infrastructure is very limited at this important level.

The next level provides infrastructure for solving simulation models associated with applications. At this level, much work has already been done on developing numerical solution libraries. Hence, our work at this level has been and will be at the meta- level: We attempt to encapsulate multiple algorithms into polyalgorithms and also integrate reasoning techniques to produce intelligent assistance.

Application area specific problem solving environments comprise the next layer of SoftLab. At this level, we are developing user-interfaces, knowledge bases, (parallel) numerical solvers, etc., that are customized for the particular application at hand.

The highest level consists of applications that are supported by adapting the application area specific problem solving environments (or *workbenches*) to the needs of the particular problem being studied. This level involves the development of very high level programming tools or composition tools that allow application scientists to combine existing tools to develop a virtual software environment that applies to the problem at hand. This is programming in the large.

In this paper we report on the developments of three prototype PSEs that try to capture the data and knowledge interactions in three "physical" labs at Purdue.

3. Computational Chromotography and Biology Softlab: SoftBIOLab

Chromatography is the process of separating components by passing a solution mixture through an absorbent column so that each component adsorbs to the surface differently than the others, and thus eludes at different times[17]. The process is used for a final purification of proteins, chemicals and biochemicals used for the manufacturing of pharmaceutical and food products, water treatment and many other bio-chemical processes. Chromatography of biological molecules is often quite complex, and many of the dynamic aspects are not well understood. Preparative chromatography uses high concentration with many solutes in the feed which introduces interference (solute competition for sorbent sites) and non-linear isotherm effects. Some components involved in a system may react with themselves (homo-aggregation) or each other (hetero-aggregation), which may result in the generation of new species. Also, reactions such as pH changes are often used to aid in the separation of biochemicals. Some solutes also undergo reactions on the solid phase, such as denaturation, or have rates of adsorption and desorption that are significantly slower than mass transfer rates, such that equilibrium isotherms are no longer valid. Any or all of these complications may be present in a given system, and each must be fully understood in order to adequately model the complete process.

To optimize output, it is generally beneficial to know precisely when and at what concentration a particular product can be found. For this purpose, the process must be mod-

eled and accurately described. With so many components and time-dependent processes occurring in chromatographic and adsorption processes, the resulting effluent history is usually difficult to predict when feed concentrations fluctuate, or when operating concentrations are changed for scale-up or optimization. Furthermore, when biological solutes are involved, experimental studies become expensive and complex. In light of those challenges, computer simulation of these complex systems becomes very attractive. Experiments are still required, but feedback from computer modeling can greatly reduce the number of experiments needed to understand and optimize the process. We are developing a problem solving environment for nonlinear bioseparation processes using the existing laboratory along with the new software and modelling tools.

3.1 The Physical Laboratory

The bioseparation laboratory operated by Linda Wang is situated in the School of

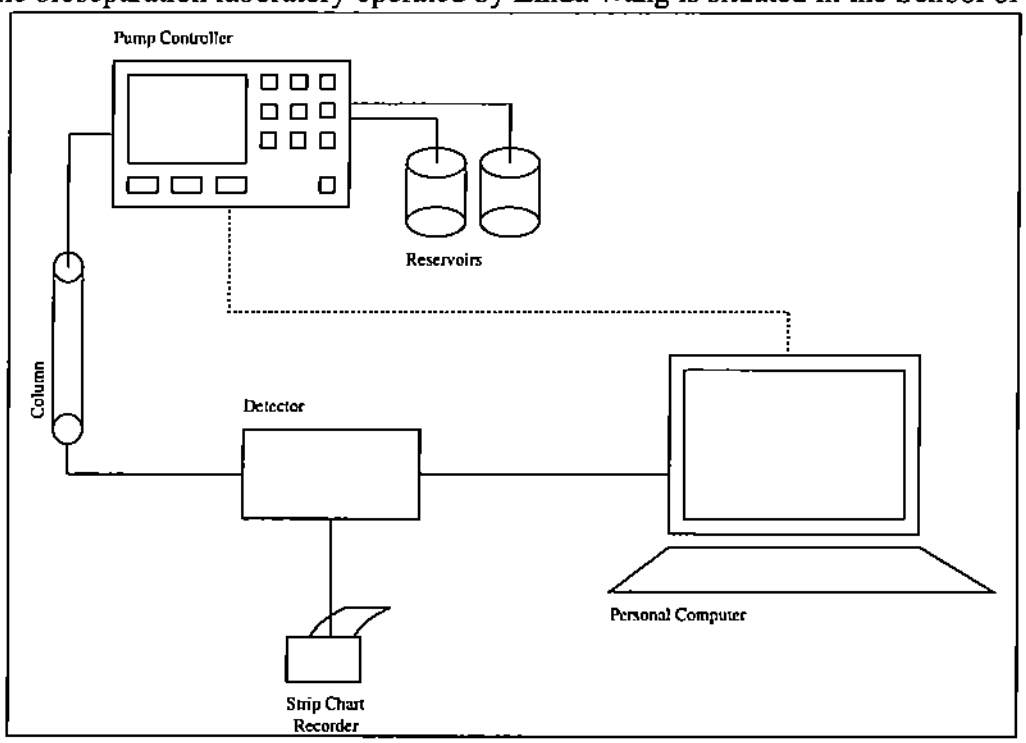

FIGURE 2. A block diagram of the equipment in the Bioseparation Laboratory in the School of Chemical Engineering.

Chemical Engineering . It now consists of three isolated computer (PCs) monitored experimental facilities and a workstation. (See Figure 2 for a block diagram view of the components in the bioseparation laboratory.)

The input solution mixture(s) are placed in reservoirs connected to the pump controller, a Waters 600E that allows the experimentor to control the mixing of the various solutes and their flow rates. The mixed solution is sent through a column where separation takes place. The output of the column is sent to the detector, a photo diode array which produces

a number of analog values representing the concentrations of various components as they elude from the column. These values are then sent to a strip chart recorder for continuous output and to a personal computer (an NEC 80286) where it is received by special hardware and processed by custom software. The custom hardware in the PC is basically an analog-to-digital convertor card and the software ("Maxima") provides convenient access to the data, including in the form of a file of numbers.

On the simulation side, the bioseparation group has a custom implementation of a numerical simulator called VERSE[21] that solves the experiment's model (partial differential) equations. Simulations are run on a Sun workstation and take on the order of hours to execute. In the current environment the interaction between the experimental instruments and the numerical models is done *manually*.

3.2 Description of the SoftBIOLab

We are approaching SoftBIOLab from two sides: First, we are interfacing the experimental instruments to the Sun workstation so as to automate the interaction between the experimental and the simulation model. Second, we are adapting their custom numerical simulator into our PDELab partial differential equation solving environment. The final step will be to interface these two components.

Our first step involves the physical interconnection of the instruments with the Sun workstation used for simulations. The industrial standard for interfacing laboratory instruments and computers is IEEE 488.2 (also known as General Purpose Interface Bus or GPIB). However, some of the instruments that are currently being used in this laboratory do not support GPIB interfaces (they use custom hardware to interface with each other and to the personal computer that gathers data). Hence, we are using an GPIB interface to the pump controller while we extract data into the workstation by using the PC as an intermediary as the detector does not support an GPIB interface. The workstation itself is on the Engineering Computer Network and hence has worldwide connectivity.

Once the instruments are physically connected to the workstation, we need to develop a software environment for setting up and controlling the experiment and for gathering results. We have implemented a prototype of this environment using the LabView[20] system. See Figure 3 for a view of the LabView interface for the bioseparation laboratory which allows us to set up some of the experiment parameters, control its workings and also to gather results via the PC. This environment also allows the user to visualize the experimental data in various ways.

On the simulation side, we have completely redesigned the //ELLPACK system[6] in order to support the diverse needs of interacting with experimental models. The new PDE solving system, called ***PDELab***, is now being implemented and a first version of the user interface environment, ***PDEView***, is operational. For simulating the experiment performed in the bioseparation laboratory, we have successfully integrated the experiment simulator software with PDELab. Our next task is to significantly improve the simulation environment by developing a parallel implementation of VERSE. Figure 4 shows a view of the PDEView interface for the bioseparation model.

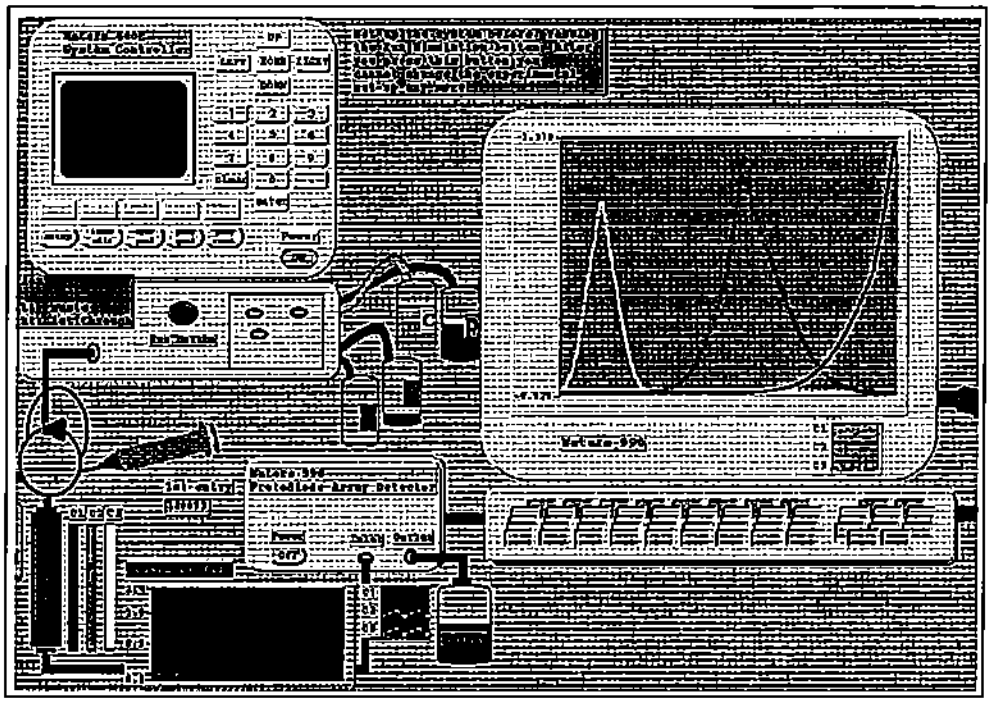

FIGURE 3. A view of the LabView interface to the instruments in the Bioseparation Laboratory.

The final step is to support the interaction of the experimental system (represented by the LabView interface) and the simulation model (represented by the PDEView interface). This integration will provide a software *workbench* consisting of various tools that support basic research in this application area. The functionality provided by this problem solving environment will therefore be:

- Collect the experimental data in real time.
- Provide visualization of the experiments.
- Control the experiment.
- Input the experimental data to the computational model.
- Control the computational model and visualize the numerical simulation solution.
- Support the parallel processing of the computational model.
- Apply various transformations on the experimental/numerical data and visualize them.
- Provide data base support for experimental/numeric data.
- Support the programming requirments of hybrid experimental/computational models.

4. Computational Electronics Softlab: SoftCELab

Epitaxial reactors are relatively high temperature, chemical vapor deposition systems used to manufacture computer chips, among other things. The demands that very large

scale integration (VLSI) and ultra LSI devices place on epitaxial systems have been the driving functions for advances in equipment technology. One such important demand is uniformity of thin film quality across a wafer and from wafer to wafer on two- and/or three-dimensional-patterned substrates. This trend is seen in several new areas of VLSI technology (as, for example, in novel three-dimensional, high performance bipolar complementary metal-oxide-semiconductors, BiCMOS). Selective Epitaxial Growth (SEG), Confined Lateral SEG (CLSEG) and Epitaxial Lateral Overgrowth (ELO) techniques allow the independent optimization of the CMOS and bipolar device parameters; yet, as minimum feature sizes shrink to smaller dimensions, knowledge and control of fundamental relationships between processing environments and material properties are lagging although they are critically important. The basic research involves the understanding of and analyses for predicting conditions at which microelectronic films can grow on two- and three-dimensional-patterned substrates. The key methodology employed consists of two parts: the use of real-time experimental observations obtained through computer-controlled chemical vapor deposition (CVD) reactors; and the use of realistic flow, temperature and concentration profiles above a susceptor, since species gas concentrations next to substrate surfaces are possibly very different from the ones in the bulk gas away from substrates.

4.1 The Physical Laboratory

The computational electronics laboratory operated by Christos Takoudis is situated in the School of Chemical Engineering. It consists of a reactor chamber that is placed inside an infrared detector connected to a computer (PC) and two controllers for controlling the temperature and pressure within the chamber. See Figure 5 for a block diagram view of the components in the computational electronics laboratory.

Data for the silicon SEG studies in CVD reactors are obtained from two experimantal systems. One of them is a CVD-Infrared (CVD-IR) reactor cell operating at pressures between 1 and 760 torr, and substrate temperatures between 25^0C and 900^0C. In this CVD-IR cell, the substrate size is typically 3-5 mm in diameter. A computer interfaced to this system is needed for mass flow control of up to four different gas flows, gas phase mass spectrometric analysis, pressure control, and substrate temperature control. This cell is already operational and used within a Nicolet 800 I IK, for in situ real-time measurements of reaction intermediates and adsorbed species during silicon epitaxial growth. The other system includes a low pressure CVD reactor which can accomodate substrate wafers of sizes up to four inches in diameter and it operates at pressures below 2 torr. A computer interfaced to this system will allow mass spectrometric analysis of the gas phase composition, pressure control and substrate temperature control. Since an interplay between experimental observations and modeling predictions (discussed next) is critical in establishing fundamental relationships between material properties and processing environments in microelectronic thin film growth on pattemed substrates, a computer system is needed for real-time measurements of selective epitaxial growth of silicon in chemical vapor deposition reactors.

To support this research we use two computer codes. One is a combination of a modification of the finite element computer code originally developed in[15] with a complex

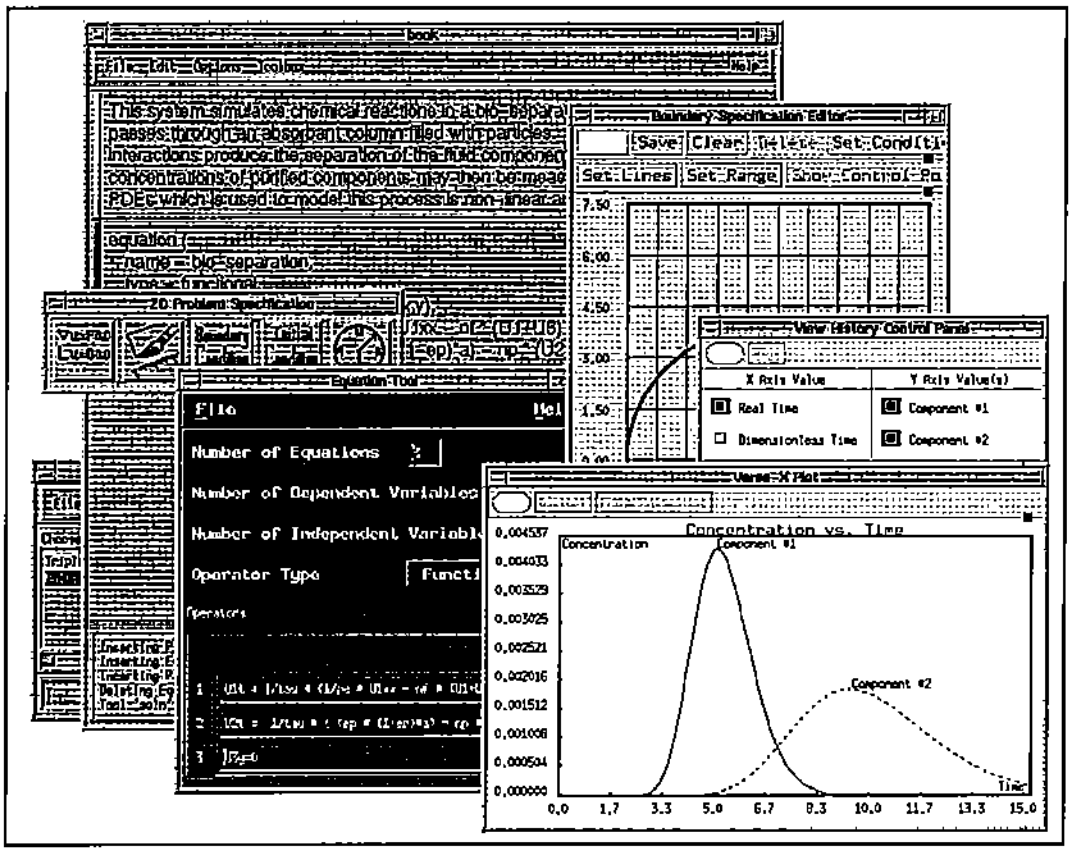

FIGURE 4. A view of the PDEView Interface for the bioseparation simulation model.

chemical equilibrium code. The other is a Petrov-Galerkin finite element code, a preliminary version of which is described in[10]; such a computer code is critical for the high linear velocities typically used in silicon SEG in CVD reactors.

The two- and axisymmetric three-dimensional codes we use demand large memory (20-40 Mbyte) and many iterative calculations because of small minimum feature sizes on patterned substrates as well as the steep velocity, concentration and temperature profiles in the CVD reaction systems proposed to be studied. Since 8-12 CPU hours are needed for a typical run of the Petrov-Galerkin finite element code, high performance computing facilities are essential in the fundamental understanding of relationships between processing environments and material properties in microelectronic thin film growth on patterned substrates with minimum feature sizes on the order of a micrometer. Coupling the experimental observations (above) with modeling simulation results is the only way to carry these studies. The SoftCELab is expected to speed these studies greatly.

4.2 Description of the SoftCELab

There are many similarities between the the chemical reaction and bioseparation laboratories. Thus the SoftCELab PSE involves the same components as the SoftBIOLab and

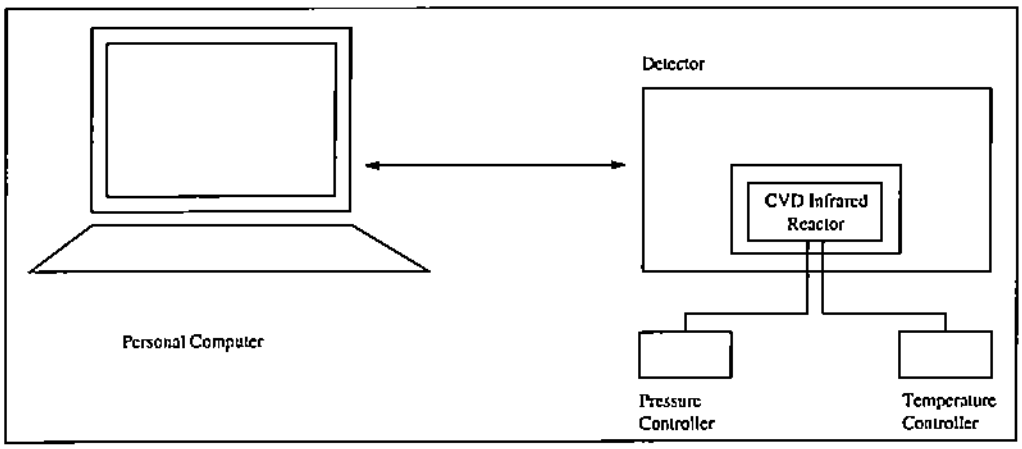

FIGURE 5. A block diagram of the equipment in the Computational Electronics Laboratory in the School of Chemical Engineering.

provides similar functionality. The difference in this instance of SoftLab is the implementation platform used. SoftCELab is supported by a QUADRA 950 system running the Mach operating system with significant multimedia support.

5. Computational Mechanics SofLlab: SoftMELab

The mission of this effort is to develop the tools and environments to study questions of mechanics through simulation. Computational mechanics involves as a significant component the inclusion of geometric models. Indeed, unless we consider only the dynamics of mass points and rigid bodies so abstracted, the interaction between the objects is modulated by their shapes, and this complicates things and necessitates geometry.

The experience with Project Newton[3][4], a rigid-body dynamics simulator driven from geometric models, suggests that geometry is a severe complication, both in terms of correctly accounting for the effects of shape on the physics in every possible configuration, as well as from a computational point of view. The effects of shape were found to be incompletely understood, or to lead outside the model domain. For example, when two knives collide with the blade edges impacting each other transversally, what is the direction of the impulsive force generated by the collision? The computational requirements dominate the simulation because geometric computations determining relative position and distance of extended complex shapes are expensive and must be repeated many times. In consequence, the initial emphasis of SoftMELab is on geometry PSEs in which the tools supporting the required geometric analyses are honed.

5.1 Description of SoftMELab

The initial project is to lay the foundation for a geometry PSE. This effort is made more difficult by the absence, in geometry, of a canonical language in which each considered shape has a unique expression. It is felt that at least a generic language should exist, in which one specifies how to construct a shape rather than what the shape is. When done

right, this generative language can be used in such a way that meaningful parts of the final shape can be named conveniently, and this should provide a formal path for information flowing from a finite element analysis of the physical properties of an object with this shape back to the generative shape description. In other words, it ought to be possible to attempt design optimization based on feedback from analysis to shape design that has been formalized, and can, therefore, be automatic. The project includes the following technical subtasks:

- *Constraint-based geometry.* Constraint examples are: two lines are parallel, a plane is incident to a point, two curves are tangent, and so on. This involves a mix of symbolic computation and numerical calculations. The key issue is how to restrict the problem, because the unrestricted problem includes geometric theorem proving.
- *Knowledge-based techniques.* Sophisticated systems infer constraints by rules such as: if two lines are nearly perpendicular, then they are in fact perpendicular. The knowledge-based component monitors inferred constraints and is capable of reproducing what constraints are in effect. When asked, it explains the constraints to the user, and seeks confirmation of constraints. Then the consequences of the user's decision has to be computed.
- *Compilation of geometric construction rules to geometric models.* Editability requires representing how to build a shape rather than the lower-level data structures of the shape itself. This is so because we have no algorithms that analyse shape for features.
- *Compilation of geometric construction rules to analysis models such as FEM.* Back feeding analysis information to shape design requires linking analysis representations to features of the shape design. Again, a high-level representation would be key.

Although the representation issue appears closely analogous to the development of a high-level programming language, there is a difference: There are two specific difficulties: People design visually, but the representation must be be digital. This entails a specific difficulty in devising a high-level representation, because it must interact perforce with the graphical user interface (UI) gestures and the specific geometric computations entailed by, e.g., selecting a face based on point-and-click. Thus, the intermediate representation communicates both with the UI and with the underlying geometry system in both directions.

Clearly, the effort must integrate foreign subsystems. They include: A geometric modeler (or several), Macsyma, a graphics subsystem, and analysis codes. In each case, suitable internal representations are key to mapping out information paths and allowing integration at the UI level.

To address the fundamental problems indicated above, we have chosen to develop a generative geometry representation without accounting for the individual capabilities and characteristics of any core geometry modeler that might be used to implement the semantics of this representation. Thus, the translation of the high-level geometry representation into specific geometry instances is analogous to the translation of a high-level programming language into machine language. The core modeling engine is then the abstract machine, while the high-level Erep is the program. As explained in[2], the analogy is in some respects inexact, but suggests the strong possiblity of federating different modelers,

and, in particular, would allow to overcome functional barriers such as the difficulty of integrating design and analysis. At the time of writing, the status of this project is as follows:

- A variational, extensible 2D geometric constraint solver is complete along with an interface that allows sketching, regeneration, as well as comprehensive recovery mechanisms in case the solver has identified a different solution than the one needed by the application. The solver is extensible.
- A pilot implementation of an Erep compiler is complete that translates and instantiates a textual geometry representation into design instances in boundary representation. ACIS is used as the core geometric modeler.
- The integration of the pilot compiler with the constraint solver and graphical design interface is in progress.

These software components are carefully insulated from underlying core modeling algorithmics. If there are dependencies, we believe they ought to be on the design interface rather than the underlying modeler, because the design interface ought to account to the user's design needs which are likely to be more permanent than the individual operations that current modeling technology can provide.

The coming steps in the project include better 3D constraint capabilities, experimenting with different mechanisms for computing persistent IDs, and new methods for interacting with the user when different design instances are to be explored.

6. Acknowledgements

The work of Elias N. Houstis, John R. Rice, Margaret Gaitatzes, and Sanjiva Weerawarana was partially funded by AFOSR grant F49620-92-J-0069 and NSF grant 9202536-CCR. The work of Ann Christine Catlin, the equipment and other support for the project was provided by NSF grant 9123502-CDA and Purdue University.

The work of Christoph M. Hoffman was partially funded by ONR Contract N00014-90-J-1599 and by NSF grant ECD 88-03017.

7. References

[1] N. P. Chrisochoides, C. E. Houstis, E. N. Houstis, P. N. Papachiou, S. K. Kortesis and J. R. Rice, "Domain decomposer: A software tool for mapping PDE computations to parallel architectures" in: R. Glowinski et. al. eds., *Domain Decomposition Methods for Partial Differential Equations IV,* SIAM Publications, 1991, pp. 341-357.

[2] C. Hoffmann and R. Juan, "EREP, an editable high-level representation for geometric design and analysis", in: P. Wilson and M. Wozny, eds., *Geometric and Product Modeling,* North Holland, 1993.

[3] C. M. Hoffmann and J. Hopcroft, "Simulation of Physical Systems from Geometric Models," *IEEE J. Robotics and Aut.*, RA3-3, 1987, pp. 194-206.

[4] C. Hoffmann and J. Hopcroft, "Model Generation and Modification for Dynamic Systems from Geometric Data," in: B. Ravani ed. *CAD Based Programming for Sensory Robots*, Springer Verlag, NATO ASI Series, F-50, 1988, pp. 481-492.

[5] C. Hoffmann and G. Vanecek, "Fundamental Techniques for Geometric and Solid Modeling" in: C. T. Leondes, ed., *Advances in Control and Dynamics*, 48, Academic Press, 1991, pp. 101-166.

[6] E. N. Houstis, J. R. Rice, N. P. Chrisochoides, H. C. Karanthanasis, P. N. Papachiou, M. K. Samartzis, E. A. Vavalis, K. Y. Wang and S. Weerawarana, "//ELLPACK: A numerical simulation programming environment for parallel MIMD machines", *Supercomputing '90*, ACM Press, New York, 1990, pp. 96-107.

[7] E. N. Houstis, J. R. Rice and R. Vichnevetsky eds., *Intelligent mathematical software systems*, North Holland, 1990.

[8] E. N. Houstis, J. R. Rice, C. E. Houstis and T. S. Papatheodorou, "ATHENA: A knowledge based system for parallel ELLPACK", in: E. Diday and Y. Lechevallier eds., *Symbolic - Numeric Data Analysis and Learning*, Nova Science, New York, 1991, pp. 459-467.

[9] I-H. Oh and C.G. Takoudis, "Modeling of Chemical Vapor Depostion in Pancake Reactor Systems," *J. Electrochem. Soc.*, 90-12, 1990, pp. 75-81.

[10] I-H. Oh, C.G. Takoudis and G.W. Neudeck, "Mathematical Modeling of the Epitaxial Silicon Growth in Pancake Chemical Vapor Deposition Reactors," *J. Electrochem. Soc. 138*, 1990, pp. 554-567.

[11] I-H. Oh and C.G. Takoudis, "Modeling of Epitaxial Silicon Growth from the SiH2Cl2-H2-HCl System in an RF-Heated Pancake Reactor," *J. Appl. Phys. 69*, 1991, pp. 8336-8345.

[12] F. Preparata and M. Shamos, *Computational Geometry*, Springer Verlag, 1985.

[13] J. R. Rice and R. Boisvert, *Solving Elliptic Problems with ELLPACK*, Springer-Verlag, New York, 1985.

[14] C. G. Takoudis and M. Kastelic, "Selective Epitaxial Growth of Silicon in a Barrel Reactor," *Chem. Engng. Sci. 44*, 1989, pp. 2049-2062.

[15] C. Taylor and T.G. Hughes, *Finite Element Programming of the Navier-Stokes Equations*, Pineridge Press, Swansea, U.K., 1981.

[16] G. Vanecek, "A Data Structure for Analyzing Collisions of Moving Objects," *IEEE Proceedings of the Twenty-Fourth Annual Hawaii International Conference on System Sciences, Volume I*, 1991, pp. 671-680.

[17] 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. 1*, 1991, pp. 142-152.

[18] S. Weerawarana and P. S. Wang, "A Portable Code Generator for CRAY Fortran", *ACM Transactions on Mathematical Software,* 18(3), 1992, pp. 241-255.

[19] S. Weerawarana, E. N. Houstis and J. R. Rice, "An interactive symbolic-numeric Interface to parallel ELLPACK for building general PDE solvers", in: B. R. Donald, D. Kapur and J. L. Mundy, eds., *Symbolic and Numeric Computing for Artificial Intelligence,* Academic Press, 1992, pp. 303-321.

[20] National Instruments Coporation, *LabView User Manual*, Autin, Texas, 1992.

[21] J. A. Berninger, R. D. Whitley, X. Zhang, N.-H. L. Wang, "A Verstile Model for Simulation of Reaction and Nonequilibrium Dynamics in Multicomponent Fixed-Bed Adsorption Processes," *Computers in Chemical Engineering,* 15, 11, 1991, pp. 749-768.