How Application Domains Define Requirements for the Grid

Inspired and guided by the limitations of scientific applications, new enabling high-performance technology is being developed for scientists and engineers, explored here through its value in chemical engineering.

Gregory J. McRae

WITHOUT DOUBT, HIGH-PERFORMANCE COMPUTING HAS TRANSFORMED THE PRACTICE OF SCIENCE AND ENGINEERING AT ALL LEVELS-A REVOLUTION THAT IS ONLY JUST BEGINNING. CONTINUING ADVANCES IN COMPUTER PERFORMANCE, SOFTWARE ENGINEERING, AND COMMUNICATION NOT ONLY WILL IMPROVE THE EFFICIENCY WITH WHICH EXISTING PROBLEMS CAN BE SOLVED BUT WILL OPEN WHOLE NEW VISTAS. A PRIMARY CHALLENGE, AND A MAJOR DRIVING FORCE BEHIND THE NATIONAL COMPUTATIONAL SCIENCE

Alliance, a participant in the National Science Foundation Partnerships for an Advanced Computational Infrastructure (PACI) program, is to realize this computational capability, not just for "expert" users but for the computational science and engineering community at large.

At the dawn of digital computing, application scientists were intimately involved in the design and programming of computational environments-an arrangement not generally in effect today. This lack of involvement is unfortunate, because application researchers and computer scientists have much to learn from one another. In the NSF Supercomputer Centers program, 1985–1997, some similar but ad hoc work was carried out by a few remote supercomputer users. But no overall effort was funded to assign computational scientists and engineers and to developing new computational infrastructure. The Alliance, beginning Oct. 1, 1997, changed that situation in the form of its six Application Technologies (see the first sidebar, "Alliance AT teams"). Particular attention is given to the teams' role in defining new enabling technologies and how the Alliance's Enabling Technologies (ET) teams can help the broader scientific and engineering com-

Alliance AT Teams

Chemical Engineering. A key goal of the Chemical Engineering team is to develop a* integrated, Web-based problem-solving environment facilitating access to high-performance computing and distributed databases and enabling interactive collaboration among team members at different sites. A particular feature of the proposed Chemical Engineer's Workbench is its ability to simplify use of multiple commercial/research codes to aid new process and product design in the microelectronics, pharmaceutical, and process-engineering industries.

Cosmology The Cosmology team's goal is to develop advanced computational tools to help astronomers understand the origin of cosmological large-scale structure as well as the formation and evolution of its astrophysical components (galaxies, quasars. and clusters). It will develop sophisticated simulation codes and analysis software. optimized for Distributed Shared Memory. that will be made available to the astronomical community.

Environmental Hydrology. Large-scale environmental processes are linked together in immensely complex systems. such as the Mississippi River basin and the Chesapeake Bay watershed. The goal of the Environmental Hydrology team is to create numerical modeling, data analysis. and visualization software that can be used in event prediction and in evaluation of environmental phenomena, with application to emergency flood management, logistical operations. and ecosystem management. It will create a* improved modeling capability by linking isolated models (atmospheric precipitation, surface water flow. watershed ecosystems) into a computational infrastructure for event-driven, multispatial, multitemporal environmental problems.

Molecular Biology. Biology is rapidly becoming an information-driven science. In recent years, technological advances in such areas as molecular genetics, protein sequencing, and macromolecular structure determination have created a* enormous flood of data. The Molecular Biology team will integrate software tools for searching and analyzing genome and protein databases for such interactive computation as visualization and docking, scientific instrument control, and molecular simulation programs (molecular dynamics, Brownian dynamics, and Monte Carlo calculations)—all from the Web-based distributed computing infrastructure provided by the *Biology* Workbench.

Nanomaterials. The Nanomaterials team will develop and apply new software tools to problems at the forefront of materials research, such as surface phenomena, cluster formation, new materials. 3D modeling and design of integrated circuits at the "deep submicron" scale. and coupling to key experimental probes. such as the scanning tunneling microscope. The Nanoelectronics subteam plans a parallel, shared computational prototyping environment for electronic device design, along with working prototypes for tools in support of computational microsystem prototyping.

Scientific Instrumentation. The Scientific Instrumentation team focuses on the problems of importing huge volumes of often time-critical observed data into high-performance computing environments for processing. Two subteams—Biological Imaging and Radio Astronomical Imaging—will each work with the ET teams to over-come the limitations of present analysis and data-handling environments by integrating imaging systems with high-performance computing environments. Together with the Cosmology team, these teams have identified the need for computational observatories and interactive, collaborative environments for the processing, display, analysis, and archiving of scientific data arising from large-scale simulations and remote instruments.

munities learn about, then embed, advances in the computing and communication infrastructure into their research programs.'

To illustrate the cross-disciplinary philosophy inherent in all Alliance activities, this article explores the approach being adopted by the Chemical Engineering team. (See Ostriker and Norman in this issue for mote on the infrastructure needs for solving very-large-scale problems in cosmology and the Cosmology team's five-year development goals.)

Applications: Driving Force for Change

A key motivation behind the Alliance's formation was the view that if discipline-specific computational scientists and engineers and computer scientists could be systematically brought together, there would be synergistic benefits to the general community. For example, application researchers collaborating with computer scientists would be able to learn about existing computational tools, tools soon to be available, and, mote important, help develop new tools that don't yet exist. Similarly, computer scientists would be able to test new ideas, algorithms, and hardware against real-world problems. The give and take between the two communities has worked well in isolated cases in the past; the next stage is to reap the benefits of a much larger U.S. collaboration.

The funded Alliance computer scientists were chosen from among leading university researchers so their areas of expertise span the hardware and software technologies needed to create a qualitatively new level of U.S. computational infrastructure. Their five-year plans are laid out in the several articles in this issue on the ET teams-Parallel Computing, Distributed Computing, and Data and Collaboration.

The AT teams were chosen to include application drivers for the numerous subcomponents of a national-scale computational infrastructure, and for their ability to build on Alliance partner strengths developed over the past decade. Selection of team members was based on several personal criteria: experience in high-performance computing and communications, participation in well-funded basic research programs in the underlying science or engineering, a willingness to disseminate their findings to a larger community, and most important, a commitment to interact with Alliance ET teams in the design, development, and rapid prototyping of an advanced computational infrastructure. As early adopters of new infrastructure, AT teams will challenge existing paradigms and drive development of innovative software tools and collaborative problem-solving environments. The span of disciplines guarantees the infrastructure developed will be generic enough to be of use to the entire research community.

Community Spirit

Is the Alliance more than just the sum of its parts? Even before funding for the Alliance became available, several all-hands meetings of the participants were held. These meetings demonstrated an emerging community spirit and a widespread willingness to collaborate. Each AT team, as well as the Education, Outreach, and Training team, prepared an initial statement describing its software needs and development plans. These statements formed the basis for iterative refinement with the ET teams' plans. The resulting sense of community and dialog were extremely encouraging. In the future, face-to-face meetings will be supplemented with electronic collaborative tools and Web-based repositories of programs, plans, and results from each project team. A particularly important result was a list of software/hardware needs common across all applications. From these needs, the three ET teams were organized:

Tools for parallel computing. All AT teams requite orders-of-magnitude increases in computing, memory, and input/output. Achieving this technology support requires better software tools for parallelizing and optimally tuning codes to architectures. Portability and scalability are more important than ever, due to the increasing use of multiple-architecture leading-edge and midrange machines and their coupling to very powerful local workstations. For example, closely coupling parallel graphics engines to parallel computers (called visual supercomputing) significantly enhances a researcher's ability to analyze the results of large computations. These projects will enhance the capabilities of the super-nodes of the National Technology Grid (see Stevens et al. in this issue).

Tools for distributed computing. Special software is needed to permit the agile scheduling and facile utilization of distributed heterogeneous computational resources for very-large-scale distributed simulations and for distributed data acquisition. Making distributed systems usable requires tools supporting the development, profiling, and optimization of distributed applications. Coupling virtual environment devices with high-bandwidth networking will allow interactive steering of algorithms and remote use of scientific instruments.

[1]This work is supported by the NSF's PACI, effective October 1, 1997

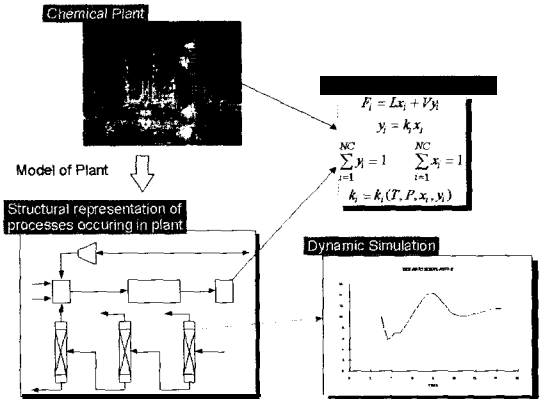

Figure I.

Predicting the chemical dynamics occurring inside a chemical plant involves modeling individual pieces of equipment as well as how the component parts interact.

Such U.S. system integration and optimization problems are far beyond the ability of any single application researcher to achieve. These projects unite the Grid's components.

Tools for data-intensive computing and collaboration. As computational science and engineering become more information-intensive, advanced technologies must be provided for efficient data storage/retrieval in a distributed environment invisible to the user. Mechanisms are required to supplement data with metadata that can be searched and correlated by distributed clients. Multiple data types have to be equally accessible, and novel scalable data visualization and **data** mining techniques have to become widely available. Technologies supporting distributed collaborative teams of developers, researchers, and educators are also needed to provide a uniform level of access to information and tools, regardless of team members' geographical locations. As computational science and engineering groups come to rely on these new methods, including immersive collaboration, session-capture and replay technologies have to be developed through distributed file and data structures. These projects add a layer to the Grid of data management, visualization, and collaborative tools, preparing the Grid for end-user applications.

A particularly striking feature of the discussions among AT teams was the frequent reference to the need for common tools. While the AT teams serve very different disciplinary communities, their software needs overlap in many areas, such as adaptive-mesh techniques for resolving dynamically evolving concentration gradients; managing large datasers; and assimilating data from remote sensors. Because the needs are common, it is possible to substantially leverage Alliance resources and bring many different perspectives to bear on problem solving.

Industrial and Community Outreach

Technology transfer is another critical aspect of the missions of all AT teams; indeed, the Alliance's overall performance will be measured by how widely the computational infrastructure it develops is adopted. Drawing on the experience of its academic partners and with input from industry, the Alliance has adopted a simplified conceptual framework for technology transfer consisting of three major stages—research, development, and deployment. The first is where ideas are transformed into research prototypes, an arena that has been well funded for several decades in universities by such federal agencies as NSF and DARPA and that has involved traditional peer-reviewed individual researchers and is the basis for U.S. technology innovation.

A **key aspect** of Alliance philosophy is to systematically build on that traditional base by funding a new infrastructure-development stage to harvest promising new research prototypes, moving them to advanced-deployment prototypes through national-scale Alliance testbeds. This complex activity requires interactions between computer science researchers, Alliance staff, and computer, communication, and software vendors. This stage was previously not systematically funded by NSF and DARPA and is precisely the reason for funding the new PACI program. The NSF review process specifically stipulates that the AT teams are not funded for basic disciplinary research but to support development of new infrastructure customized to the requirements of their respective disciplines. Within the Alliance framework, each team's performance will be peer-reviewed and evaluated annually while providing an opportunity to introduce new application areas.

The last stage—infrastructure deployment—is mainly carried out by the private sector. Through various mechanisms, the private sector chooses the infrastructure prototypes most worthy of its major investment and support. The Alliance has selected a set of vendors—including Silicon Graphics/Cray Research, Hewlett-Packard, IBM, Sun Microsystems, Microsoft, Computer Associates, Ameritech, MCI, and AT&T—to help support and evaluate many Alliance infrastructure development projects.

The Alliance can also participate through its Education, Outreach, and Training team, helping build a new constituency of users who would benefit from access to the new infrastructure. The NCSA Industrial Partners are also closely linked to this process and, through their internal use of the advanced-deployment prototypes, can themselves generate demand for the new infrastructure.

Chemical Engineering/Alliance Interaction

One of the AT teams—Chemical Engineering—illustrates the Alliance's general approach toward interaction between each team. The explosive growth in computational power available to the chemical engineering professions has been exploited in a variety of practical applications, from molecular modeling to the simulation of the operation of complete production plants. Continuing advances in supercomputer speed and memory size, as well as ever-expanding networking and communication infrastructure, now influence every aspect of chemical engineering practice, from product and process design to crisis management and from everyday administrative activities to management of corporate information flow on a global scale [3, 4]. As in all the AT teams, the critical issue is how to continue to exploit the emerging computational, communication, and information capabilities.

One chemical engineering subdiscipline—process modeling—has traditionally used high-performance computing for flow-sheet simulations, in which mathematical models are used to describe the interactions among various chemical processes occurring inside a plant. These models are typically very large systems of nonlinear algebraic or differential equations derived from descriptions of the processing units in the plant, their interconnection topology, and their design specifications. An illustration of this idea is shown in Figure 1 and described in more detail in [1]. In practical applications, a whole plant simulation might involve more than a million equations exhibiting a wide range of characteristic time scales. These models are of enormous economic importance to the design and operation of large plants.

Just solving a flow-sheet model can tax the resources of even the largest computers. When the models are used for optimal design calculations, in which many thousands of alternatives might be considered, or when real-time-response solutions are needed for process-control applications, the demand for computational power escalates rapidly. Chemical Engineering team member Richard Braatz from the University of Illinois at Urbana-Champaign is involved in developing plant-wide control systems through assimilation of very large quantities of live process data from sensors and instruments, rapid control calculations, and real-time response. So, in addition to more raw computer power, researchers need the ability to manipulate very large datasets. Tight integration of controls, process data, and

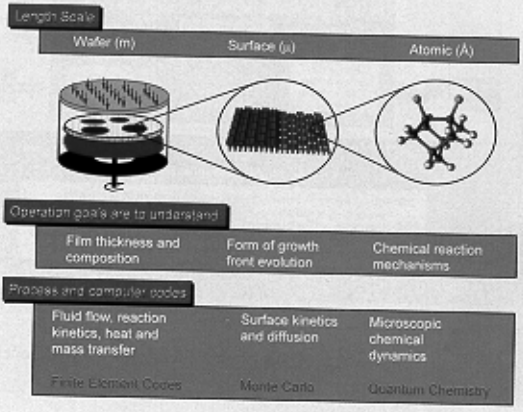

Figure 2.
Semiconductor manufacturers need to understand the physical and chemical processes leading to improved wafer yields and surface morphologies. Models and the associated solution algorithms have to scale from the atomic to the dimensions of the chemical reactor where the wafers are manufactured. (Based on the work of Chemical Engineering team member Klavs Jensen at MIT.)

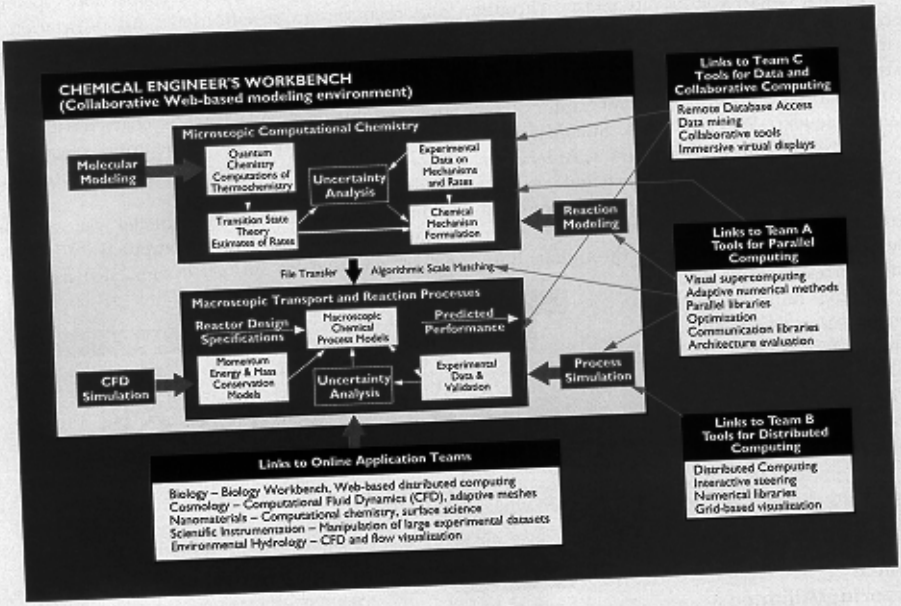

Figure 3.
Basic components of the Chemical Engineer's Workbench, with its links to Alliance teams

process modeling is critical for completing the feedback loop needed to keep a chemical plant operating optimally.

While problem size and data management are clearly important issues, often overlooked is that the time needed to formulate and build the models themselves may be even more expensive. Thus, there is also a critical need for better modeling environments [4].

The Chemical Engineering team will work closely with the ET teams to achieve its primary five-year goal: developing an integrated Grid environment for chemical process modeling, control, and optimization that runs in a portable fashion on scalable computers with tight coupling to an efficient distributed file and data infrastructure. As a starting point, the Advanced Batch and Continuous Unsteady-State Simulator (ABACUSS) software system, developed by Paul Barton at MIT, will be used to test the enabling

Table 1.
Key infrastructure requirements to support flow-sheet modeling, excerpted from a longer discussion of the needs, schedules, and project described in the Alliance's proposal to NSF [6]. (Note: A = Parallel Computing team; B = Distributed Computing team; C = Data and Collaboration team.)

Algorithmic and Infrastructure Needs to Support ABACUSS	
Problem Areas	**Links to Enabling Technologies Teams (ABACUSS) and other Applications Areas**
Linear Algebra	Scientific libraries for solving linear systems of equations on parallel and distributed machines (A, B).
Nonlinear Equations	Parallel extensions of automatic differentiation techniques for simultaneous residual and Jacobian evaluations in Newton's algorithm (A, B).
Process Optimization	Parallel methods for sequential quadratic programming on distributed shared memory machines (A).
Data Assimilation and Process Control	High-performance storage and information management (C); Scientific Instrumentation AT team.
Solution of Partial Differential Equations	Algorithms for adaptive meshing (A); Cosmology and Environmental Hydrology AT teams.
Visual Analysis	Desktop 3D virtual environment tools (C).
Web-based Collaborative Modeling Environments	Distributed data storage and integrated heterogeneous data storage across networks (B, C); Molecular Biology AT team.

technologies for solving large sets of equations (see Table 1). ABACUSS is an ideal test environment because it incorporates many software innovations, including object-oriented technology for modeling chemical processes, as well as dynamic simulation and the ability to use mixed continuous/discrete representations of process variables. The system is also capable of addressing industrial-strength problems of the type supplied as test cases by industries.

The traditional chemical engineering approach—choosing the process chemistry first, then developing the manufacturing system-has resulted in high costs for satisfying environmental regulations. The key to the design of clean, efficient, and profitable manufacturing industries is to integrate the selection and application of product chemistry processing technologies, optimizing economic performance and environmental quality. Furthermore, multiscale models that depend on adaptive methods are needed to introduce realism into chemical plant computations.

The ability to quickly solve the plant model sets the stage for optimizing overall system performance. The MIT group will work with the Parallel Computing team to develop parallel techniques to optimize multistage dynamic systems, dynamic systems with integer decisions, and dynamic optimization of partial differential equation systems, all built on the ABACUSS platform.

Chemical Engineering and the Next Generation of Microelectronics

Microelectronics are an integral and dynamic part of modern society. The trends toward equipment supporting 300mm-wafer production, features smaller than 0.2 micron environmentally friendly process chemistries, and faster development times have created a pressing need to better understand how to design and operate production facilities. It is ironic that John von Neumann, originally trained as a chemical engineer, recognized early the potential benefits of massive parallelism and the barriers posed by a limited understanding and control of the fabrication processes for circuit elements (see Figure 2).

Finding either a good design or the best set of operating conditions for the process engineering of a microelectronic device is a challenging task that has to balance the complex relationships among film growth, yield, and production times. Today, this task is frequently done through large and costly experimenration. The microelectronics industry is currently involved in a major set of development activities designed to produce the kind of process simulation tools that have been available in the mote traditional process industries. While there ate many similarities to the problems described earlier, there are also some new aspects. One of the most important is the treatment of the very wide variation in space scales that have to be considered for even an approximate model.

The Nanomaterials team will work on the algorithms and applications needed to determine the physical and electronic properties of the electronic devices themselves. The Chemical Engineering team and the Nanomaterials team will ultimately explore novel surface chemistry phenomenon on ultra-small-scale electronic devices and the processes needed to manufacture them efficiently [5].

The challenge is to take the outputs from models of processes occurring on one scale and use them as inputs to the next level. The problem is further complicated by the fact that the modeling tools are often proprietary computer codes designed to operate at only one of the needed scales. The Chemical Engineering team will collaborate with the Data and Collaboration team to design data exchange standards for information movement among codes and with the Distributed Computing team to implement a solution manager able to run the individual codes in a distributed computing environment.

In addition to the scale-matching issue, in which different codes have to be linked together for a complete simulation, another problem-horizontal code integration-often occurs in process engineering calculations. Most process modeling codes make extensive use of physical property data. In practice, this information can be determined by direct measurement, from literature searches, or through different estimation methodologies that can vary from simple empirical formulae to complex quantum chemistry calculations.

Depending on which approach is used, there may be large uncertainties in the numbers input into the process modeling codes. The Chemical Engineering and Data and Collaboration teams will jointly develop new database representation schemes that identify the different sources of information, remember the processing pathways used, and keep track of the estimates of the uncertainties. New algorithms for propagating uncertainty that can exploit parallel and distributed architectures will be developed in collaboration with the Parallel Computing and Distributed Computing teams. Many of the issues involved in managing multiple data sources in process engineering are the same as those encountered by the Molecular Biology team in the Biology

Biology Workbench

In order to create an integrated computational environment to make molecular biology databases interoperable with each other and with programs for analysis, visualization, and simulation, NCSA's Computational Biology Group developed the Biology Workbench (see http://bioweb.ncsa.uiuc.edu/), consisting of a Web browser user interface and a Web server connected to an evolving collection of biological sequence and structure analysis tools and easy-access links to large biological sequence and structure databases.

Previously, biologists had to run specific software packages on their desktop computers and gather datasets by hand from carefully maintained databases. The packages were often not developed for cross-platform use and could be run only from the desktop on which the software was installed. In contrast, the Biology Workbench provides a universally accessible framework that can be viewed from any Web browser on any Internet-connected computer. The programs and distributed data are gathered together onto the NCSA Biology Workbench server and updated daily. Moving data from one task to another is accomplished by on-the-fly translation library technology developed at NCSA. A researcher knows only that a complete set of computational tools and worldwide distributed databases is accessible from any computer with a Web browser.

The Molecular Biology team, which spans many major U.S. groups and centers, will develop and integrate software within the Web-based distributed computing infrastructure provided by the Biology Workbench. The internationally recognized leaders in genomics, bioinformatics, computational molecular biology, and Web-based distributed computing who are members of this team will add capabilities for:

- Large-scale human genome sequencing studies
- A microbial ribosomal database for studying the comparative evolution of life
- An x-ray structure solving environment
- A complete computational environment linked to the Workbench for obtaining nuclear magnetic resonance structures
- State-of-the-art molecular biology computation tools for modeling diffusion of biological macromolecules, electrostatic interactions between macromolecules, and molecular dynamics of macromolecules
- An environment for visualization and analysis of proteins

The current Web-based Biology Workbench will be enhanced by the addition of collaborative capabilities using NCSA Habanero. Visualization capabilities will be enhanced by Java applets. The "BioWidgets" project is dedicated to producing reusable Java display tools for studying biological data types. A general workbench schematic is shown in the figure. Note that the software architecture (Web browser, tools, and data) is general and will be adopted by the other AT teams.

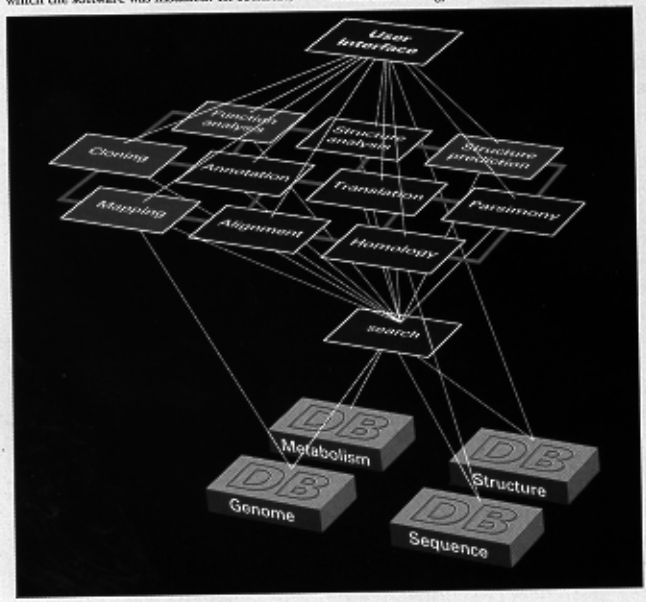

Figure. The server-based Workbench query engine/program driver launches queries to remote or local databases and information sources, returns the answers to the query engine/program driver, directs the answers either to the user's workstation or into input files of remote or local applications, launches these applications, receives the application results, and directs these results to the user's workstation.

Workbench (see the second sidebar, "Biology Workbench").

The Chemical Engineer's Workbench

Historically, most computational science and engineering data was generated, processed, and analyzed by a single researcher or a tightly knit research group located at a single site. Cutting-edge computational science and engineering increasingly requires distributed access to very large data archives, sophisticated information mining, visualization techniques, and collaborative exploration and data analysis. The overall Grid integration framework for the Chemical Engineering team will be the Chemical Engineer's Workbench, a Web-based computational environment for process engineering; it will be a unified, multiscale environment for both reactor engineering and optimization, extensible for higher levels of integration.

A schematic representation of the elements of the Chemical Engineer's Workbench is shown in Figure 3, which also shows the interlocking ties to the other Alliance teams involved in bringing the full Workbench into being. Perhaps in this example one can best see the reasoning behind building the Alliance from these teams, which provide specialized knowledge both in computer science disciplines and in customizations of these disciplines to diverse computational science and engineering fields. Because the Alliance primarily funds the people who "glue" ream researchers together, projects like the Workbench are the driving applications guaranteeing constant interaction across disciplinary boundaries.

To address the difficulties inherent in integrating across scales and disciplines, the Chemical Engineering team will build synchronous collaborative environments around intersections of disciplines (for example, spanning the gap between quantum chemistry calculations and macroscopic chemical process modeling calculations). The actual linkage of the diverse computational paradigms in the Chemical Engineer's Workbench will be accomplished through a combination of Web programming (much as in the Biology Workbench) and NCSA's Habanero [2] or Syracuse University's Tango synchronous collaboration tools (see Reed et al. in this issue).

The initial Workbench prototypes will link only a few component parts. As each of them matures, the Chemical Engineering team will work closely with the Education, Training, and Outreach team to release the Workbench for use in classroom education. For example, undergraduate chemical engineering students will have access to the best chemical process engineering tools, linked in a seamless and logical fashion to the multiscale, multidisciplinary aspects of chemical engineering. The Workbench will serve as a framework for demonstrating how to perform a variety of real-world chemical engineering computations. In particular, it will allow students to move beyond the complexity of numerical implementations and focus instead on the details of the engineering problem.

Conclusions

A primary Alliance goal is to provide desktop access to the Grid-the most powerful computational science and engineering infrastructure ever assembled. In the case of chemical engineering, this high-performance computing environment will make it possible to design chemical processes and products that are commercially successful and improve environmental quality--a prospect both exciting and economically important to the future of the chemical-engineering profession throughout the world. By embracing the computational challenges arising from many different problem domains, Alliance computer scientists, engineers, and application researchers can help guide and shape the course of the information revolution together. C

References

1. Barton, P., and Pantelides, C. Modeling of combined discrete/continuous processes. *AIChE J. 40*, 6 (1994), 966–979.
2. Driggers, B., Alameda, J., and Bishop, K. Distributed collaboration for engineering and scientific applications implemented in Habanero, a Java-based environment (see http://www.ncsa.uiuc.edu/SDG/Software/Habanero/HabaneroHome.html).
3. Koch, D. The future: Benefiting from new tools, techniques, and teaching. *Chem. Eng. Prog.* (Jan. 1997), 66–72.
4. Krieger, J., and Freemantle, M. ACHEMA 97: The future on display. *Chem. Eng. News 75*, 27 (1997), 10–21.
5. Simka, H., Hierlemann, M., Utz, M., and Jensen, K. Computational chemistry predictions of kinetics and major reaction pathways of germane gas-phase reactions. *J. Electrochem. Soc. 143* (1996), 2646–2654.
6. Smarr, L., et al. National Computational Science Alliance: Empowering Computational Science, Engineering, and Education through Partnerships for an Advanced Computational Infrastructure. Proposal to the U.S. National Science Foundation, Washington, D.C., 1996.

GREGORY J. MCRAE (mcrae@mit.edu) is is the Bayer Professor of Chemical Engineering at the Massachusetts Institute of Technology, a member of the Alliance Executive Committee, leader of the Chemical Engineering team, and co-chair with Ken Kennedy of the Parallel Computing team.