Purdue University
Purdue e-Pubs

Department of Computer Science Technical Reports | Department of Computer Science

1995

MPSE: Multidisciplinary Problem Solving Environments

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

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

Anupam Joshi

Sanjiva Weerawarana

Elisha Sacks
Purdue University, eps@cs.purdue.edu

See next page for additional authors

Report Number:
95-047

Houstis, Elias N.; Rice, John R.; Joshi, Anupam; Weerawarana, Sanjiva; Sacks, Elisha; Wang, Nien-Hwa Linda; Takoudis, Christos; Sameh, Ahmed; and Gallopoulos, Efstratios, "MPSE: Multidisciplinary Problem Solving Environments" (1995). *Department of Computer Science Technical Reports.* Paper 1222.
https://docs.lib.purdue.edu/cstech/1222

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

Authors

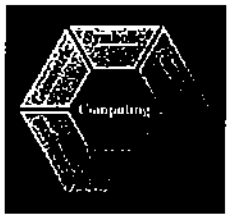
Elias N. Houstis, John R. Rice, Anupam Joshi, Sanjiva Weerawarana, Elisha Sacks, Nien-Hwa Linda Wang, Christos Takoudis, Ahmed Sameh, and Efstratios Gallopoulos

This article is available at Purdue e-Pubs: https://docs.lib.purdue.edu/cstech/1222

MPSE: Multidisciplinary Problem Solving Environments

Elias N. Houstis, John R. Rice, Anupam Joshi,
Sanjiva Weerawarana, Elisha Sacks and Vernon Rego
Nien-Hwa Linda Wang and Christos Takoudis
School of Chemical Engineering
Computer Sciences Department
Purdue University
West Lafayette, IN 47907
Ahmed H. Sameh and Efstratios Gallopoulos
Department of Computer Science
University of Minnesota

CSD-TR-95-047
July, 1995



MPSE: Multidisciplinary Problem Solving Environments

Elias N. Houstis, John R. Rice, Anupam Joshi, Sanjiva Weerawarana, Elisha Sacks and Vernon Rego

Department of Computer Sciences, Purdue University

Nien-Hwa Linda Wang and Christos Takoudis

School of Chemical Engineering, Purdue University

Ahmed H. Sameh and Efstratios Gallopoulos

Department of Computer Science, University of Minnesota

A ABSTRACT

The National Information Infrastructure (NII) that will emerge in the 1990's will impact many institutions of life; it will change the way we learn and do science, access information, work and collaborate and design physical systems. This future computational power and communication infrastructure will allow *computing everywhere*. Learning and training simulators will be part of any classroom and laboratory. The very concept of classroom, laboratory and individual workplace will change; they will become virtual places based on an array of multimedia devices. The software architecture of such an environment needs to be developed, and related research issues on *intelligent* computing, *intelligent* interfacing and *intelligent* communication, integration of computing systems (i.e., symbolic, numeric, geometric and office) on mobile platforms have to be addressed. The advent of such environments will also affect the process of *prototyping*, which 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[67]. This, in turn, requires the fast, accurate simulation of physical processes and design optimization using knowledge and computational models from *multiple disciplines* in science and engineering [32]. Thus, the realization of **rapid multidisciplinary problem solving** or **prototyping** is the new grand challenge for Computational Science and Engineering (CS&E) [68][73]. We refer to a software realization of multidisciplinary prototyping throughout as a **Multidisciplinary Problem Solving Environment** (MPSE).

We describe in this white paper research that is needed to formulate and develop a mathematical and software *framework* for MPSE including the tools, enabling technologies, and underlying theories needed to support physical prototyping in the *classroom*, *laboratory*, *desk*, and *factory*. It will utilize the current and future NII facilities and HPCC technologies. It will be *adaptable* and *intelligent* with respect to end-users and hardware platforms. It will use collaborating software systems and agent based techniques to build demonstration MPSEs for physical modeling and education. It will allow wholesale reuse of scientific software and provide a natural approach to parallel and distributed problem solving.

The evolution of the National Information Infrastructure (NII) also has the potential to revolutionize education. We envisage that universities will offer telecourses using the NII and their student body will be geographically dispersed. In this research, we also aim to build demonstration MPSEs that will provide a virtual classroom and laboratory to support such distance education.

The assumed hardware scenario for the proposed MPSE development is based on the fact that the advent of optical, ATM, and wireless hardware communication technologies and distributed computing infrastructure like the Web (WWW) will make the concept **"The Network is the Computer"** a reality [76]. We assume that **"The Net"** interconnects computational units (ranging from workstations to massively parallel machines) and physical instruments and that it supports intelligent, two-way video communication. Moreover, the WWW like infrastructure is viewed as an object-oriented operating system kernel that allows the development of an MPSE as a distributed application utilizing resources and services from many sources. The specific tasks that have to be accomplished to realize MPSEs are as follows:

Foundations of PSEs: (i) Formulation and analysis of multidisciplinary problem solving methodologies on HPC platforms, (ii) Creation of PDEPack, a broad library of PDE problem solvers and components of PDE solvers, (iii) Enhancement of the configuration space technology for mechanical systems design, (iv) Development of knowledge bases for PDE based prototyping.

Enabling Technologies for PSEs: (i) Creation of kernels for building domain specific PSEs and MPSEs on virtual parallel environments, (ii) Development of a methodology for virtual and smart libraries of problem solvers, (iii) Creation of an infrastructure to deliver MPSE problem solving power over the network.

Demonstration PSEs and MPSEs: (i) Education: *SciencePad* to support the classroom and the *"collaboratory"* of the future on ubiquitous platforms, (ii) Numerical Infrastructure: PDELab for partial differential equation based applications, (iii) Electronic prototyping: EPPOD for the design and analysis of physical systems.

In all the above task *the Net* is assumed to be the host for multidisciplinary problem solving. *The Net* will supply and support distributed applications, PSEs, libraries of solvers, distance learning and "collaboratory" services, and domain specific knowledge bases. We envision that PSEs, mathematical software systems, and courseware will take the form of special servers in the network which are constantly updated by their creators. The users will be able to utilize all available PSEs and mathematical/engineering libraries as easily as browsing a Web page today.

We will present, in the sections that follow, an overview of what we feel are the significant areas of research for HPCC in the information age. We will also describe our own plans and efforts to address some of the challenging issues that arise as high performance, scientific computing meets the information revolution.

B SCOPE AND GOALS OF RESEARCH

It is predicted that by the beginning of the next century, the available computational power will enable anyone with access to a computer to find an answer to any question that has a known or effectively computable answer. The proposed research in the area of problem solving environments (PSEs) promises to contribute toward the realization of this prediction for physical modeling and to provide students, scientists, and engineers with environments that allow them to spend more time doing science and engineering rather than "computing".

The predicted growth of computational power and network bandwidth suggests that computational modeling and experimentation will be one of the main tools in *big* and *small* science. In this scenario, computational modeling will shift from the current single physical component design to the design of a whole physical system with a large number of components that have different shapes, obey different physical laws and manufacturing constraints, and interact with each other through geometric and physical interfaces. For example, the analysis of an engine involves the domains of thermodynamics (gives the behavior of the gases in the piston-cylinder assemblies), mechanics (gives the kinematic and dynamic behaviors of pistons, links, cranks, etc.), structures (gives the stresses and strains on the parts) and geometry (gives the shape of the components and the structural constraints). The design of the engine requires that these different domain-specific analyses interact in order to find the final solution. The different domains share common parameters and interfaces

but each has its own parameters and constraints. We refer to these multi-component based physical systems as ***multidisciplinary applications*** (MAs).

The realization of the above scenario, which is expected to have significant impact in industry, education, and training, will require the development of *new algorithmic strategies* and *software* for managing the complexity and harvesting the power of the expected HPCC resources; it will require *PSE technology* to support programming-in-the-large and reduce the overhead of HPCC computing. *A goal of research in this area should be to identify the framework for the numerical simulation of multidisciplinary applications and to develop the enabling theories and technologies needed to support and realize this framework in specific applications.* The MPSE is the software implementation of this framework. It is assumed that its elements are discipline-specific problem solving environments (PSEs). The MPSE design objective is to allow the "natural" specification of multidisciplinary applications and their simulation with interacting PSEs through mathematical and software interfaces across networks of heterogeneous computational resources.

The future NII will impact many aspects of life, including the way we learn and do science, access information, work, and collaborate. It will allow *computing everywhere*[33]. Learning and training simulators will be part of every classroom and laboratory. The concept of the classroom, the laboratory, and the student/scientist/engineer desk environments will evolve to some virtual form based on an array of multimedia devices [37]. These virtual environments, sometimes called "collaboratories" [45], can be implemented naturally using the MPSE technology that will develop. We are building discipline-specific PSEs that will be used to build learning and training simulators in some areas of computational science and engineering (CS&E).

For the scope of this discussion, **the network** (*the Net*) is assumed to be the host for multidisciplinary problem solving. One can use and modify existing network software infrastructure to support distributed applications, PSEs, libraries of solvers, and distance learning and "collaboratory" services over *the Net*. We envisage that all these resources and services will take the form of special servers in the network which are constantly updated by their creators. The users will be able to utilize all available PSEs and mathematical/engineering libraries as easily as browsing a Web page today. *We feel that there is a need to create servers to experiment with web based dissemination and use of PDE software, PSEs, and courseware.*

C RESEARCH ISSUES TO BE ADDRESSED

C.1 Domain specific PSEs

Even in the early 1960s, scientists had begun to envision problem-solving computing environments not only powerful enough to solve complex problems, but also able to interact with users on human terms. The rationale of our research is that the dream of the 1960s will be the reality of the 1990s: High performance computers combined with better algorithms and better understanding of computational science have put PSEs well within our reach.

What are PSEs? A PSE is a computer system that provides all the computational facilities needed to solve a target class of problems. These facilities include advanced solution methods, automatic selection of appropriate methods, use of the application's language, selection of appropriate hardware and powerful graphics, symbolic and geometry based code generation for parallel machines, and programming-in-the-large. The *scope* of a PSE is the extent of the problem set it addresses. This scope can be very narrow, making the PSE construction very simple, but even what appears to be a modest scope can be a serious scientific challenge. For example, we propose to create a PSE for bioseparation analysis as described later. This has a narrow scope, but is still a complex challenge as we incorporate both a computational model and an experimental process supported by physical laboratory instruments. We are also creating a PSE called PDELab for partial differential equations (PDEs). This is a far more difficult area than bioseparation and the resulting PSE will be less

powerful (less able to solve all the problems posed to it), less *reliable* (less able to guarantee the correctness of results), but more *generic* (more able to "parse" the specifications of many PDE models). Nevertheless, PDELab will provide a quantum jump in the PDE solving power delivered into the hands of the working scientist and engineer.

What are the PSE related research issues to be addressed? A substantive research effort is needed to lay the foundations for building PSEs. This effort should be directed towards i) a PSE kernel for building scientific PSEs, ii) a knowledge based framework to address computational intelligence issues for PDE based PSEs, iii) infrastructure for solving PDEs, and iv) parallel PDE methodologies and virtual computational environments. The description of these proposed tasks can be found in later in this document.

C.2 MPSEs for prototyping of physical systems

If PSEs are so powerful, what then is an MPSE? In simple terms, an MPSE is a framework and software kernel for combining PSEs for tailored, flexible multidisciplinary applications. A physical system in the real world normally consists of a large number of components which have different shapes, obey different physical laws and manufacturing/design constraints, and interact through geometric and physical interfaces. Mathematically, the physical behavior of each component is modeled by a PDE or ODE system with various formulations for the geometry, PDE, ODE, interface/boundary/linkage and constraint conditions in many different geometric regions. In the case of complicated artifacts such as the automobile engine, which has literally hundreds of odd shaped parts and a dozen physical phenomena, it is difficult to imagine creating a monolithic software system to model accurately such a complicated real problem. Therefore, one needs an MPSE mathematical/software framework which, first, is applicable to a wide variety of practical problems, second, allows for software reuse in order to achieve lower costs and high quality, and, finally, is suitable for some reasonably fast numerical methods. Most physical systems and manufactured artifacts can be modeled as a ***mathematical network*** whose nodes represent the physical components in a system or artifact. Each node has a mathematical model of the physics of the component it represents and a *solver agent* [14]for its analysis. Individual components are chosen so that each node corresponds to a simple PDE or ODE problem defined on a regular geometry.

What are the mathematical network methodologies required? What are the research issues? There exist many standard, reliable PDE/ODE solvers that can be applied to these local node problems. In addition there are nodes that correspond to interfaces (e.g. ODEs, objective functions, relations, common parameters and their constraints) that model the collaborating parts in the global model. Moreover, the analysis of an artifact changes through time, thus some of the interfaces appear and disappear during the analysis session. To solve the global problem, we let these local solvers collaborate with each other to relax (i.e., resolve) the interface conditions. An interface controller or mediator agent collects boundary values, dynamic/shape coordinates, and parameters/constraints from neighboring subdomains and adjusts boundary values and dynamic/shape coordinates to better satisfy the interface conditions. Therefore, the network abstraction of a physical system or artifact allows us to build a software system which is a network of collaborating well defined numerical objects through a set of interfaces. Some of the theoretical issues of this methodology have been addressed in [52] [49] for the case of collaborating PDE models. The results obtained so far verify the feasibility and potential of network-based prototyping.

A major avenue of research is to further study the mathematical network approach in realistic applications like the two-stroke engine design. Moreover, we feel that one should explore a new MPSE methodology based on the so called prefabricated numerical objects. These approaches will be realized in the form of a software kernel that will allow to build demonstration MPSEs for physical systems by reusing "legacy" software and domain specific PSEs.

What are the software methodologies for implementing the "mathematical network"? What are the research issues? A successful architecture for PSEs requires heavy reuse of existing software within a modular, object oriented framework consisting of layers of objects. The kernel layer integrates those components common to most PSEs or MPSEs

for physical systems. We observe that this architecture can be combined with an "agent oriented paradigm" and "collaborating solvers" [14] to create MPSE as a powerful prototyping tool. The creation of these frameworks and kernels requires us to create a variety of infrastructure tools and that, plus some applications of PSEs and MPSEs, comprises the research tasks of this project. The principal applications are in chemical engineering (bioseparation of food and drug products, convective vapor deposition in micro-chip manufacturing), education (electronic classrooms and laboratories), numerical infrastructure (PDELab) and engineering design of physical systems.

MPSEs must exploit and build on the new technologies of computing. By the time MPSEs are operational, the advances in computing power and the communication infrastructure will allow ubiquitous high performance computing, i.e., every where by every one. The designs for MPSE must be application and user driven. An MPSE must simultaneously minimize the effort and maximize the solution power delivered to researchers, engineers and scientists, students, and trainees. We should not restrict our design just to use the current technology of high performance computers, powerful graphics, modular software engineering, and advanced algorithms. We should see MPSE as delivering problem solving services over *the Net*. This viewpoint leads naturally to collaborating, agent based methodologies. This, in turn, leads to very substantial advantages in both software development and quality of service as follows. We envision that a user of MPSE will receive at his location only the user interface. Thus the MPSE server will export to the user's machine an agent that provides an interactive user interface built on top of the standard services of *the Net*. The bulk of the software and computing is done at the server's site using software tailored to a known and controlled environment. The server site can, in turn, request services from specialized resources it knows, e.g., a commercial PDE solver, a proprietary optimization package, a 1000 node supercomputer, an ad hoc collection of 122 workstations, a database of physical properties of materials. Each of these resources is contacted by an agent from the MPSE with a specific request for problem solving or information service. Again, all this collaboration is built on standard Network services. All of this can be managed without involving the user (if s/he so desires), without moving software to arbitrary platforms, and without revealing source codes.

What are the design objectives of an MPSE for physical system design? What are the research issues? These mathematical networks can be very big for major applications. For a realistic vehicle simulation, there are perhaps 100 million variables and many different time scales. This problem has very complex geometry and is very non-homogeneous. The answer is 20 gigabytes in size and requires about 10 teraflops to compute. An answer is a data set that allows one to display an accurate approximate solution at any point. This data set is much smaller than the computed numerical solution. The network has 10,000 subdomains and 35,000 interfaces. A software network of this type is a natural mapping of a physical system and simulates how the real world evolves. This allows the use of the software parts technology (object-oriented programming) that is the natural evolution of the software library idea. It allows software reuse for easy software update and evolution, things that are extremely important in practice. The real world is so complicated and diverse that we believe it is impractical to build monolithic, universal solvers for such problems. Without software reuse, it is impractical for anyone to create on his own a large software system for a reasonably complicated application. Each new automobile normally results in a new software system. Recreating such a system could easily take several months or years. In contrast, the execution time to perform the required computation might only be a few days. Notice that such a physical change usually corresponds to replacing, adding, or deleting a few nodes in the network with a corresponding change in interface conditions. These are simple manipulations on a network which do not affect the rest of the system and can thus be easily done. In this application each physical component can be viewed both as a physical object and as a software object. In addition, this mathematical network approach is naturally suitable for parallel computing as it exploits the parallelism in physical systems. One can handle issues like data partition, assignment, and load balancing on the physics level using the structure of a given physical system. Synchronization and communication are controlled by the mathematical network specifications and are restricted to interfaces of subdomains, which results in a coarse-grained computational problem. This is especially suitable for today's most advanced parallel supercomputer architectures. The network approach also allows high scalability.

Realizing this MPSE technology requires research advances both in the general structure and implementation area and in more specific areas from the applications we consider. For example, we must design and create the tools that allow the

MPSE agents to collaborate over *the Net*. We must create a flexible and general methodology for interfacing large and heterogeneous software systems. We will advance the computational models currently used at CVD process and map them to proposed virtual parallel environments. For example, in the specific applications, we will define and create a library (PDEPack) of PDE solving modules including classical numerical linear algebra routines and modern distributed mesh generators or solvers. We will advance the practice of identifying high level characteristics of partial differential equations and their use for selecting solutions methods and computing platforms. We will integrate the CVD laboratory equipment with the PDELab solving power to create a unified software/hardware environment for studying micro-chip manufacturing.

C.3 "Collaboratory" MPSE for the classroom and laboratory

How are MPSEs used in the classroom? The PSEs and MPSEs provide high level systems and models for physical processes. Students can manipulate realistic models by simulation, or even actual physical apparatus in the laboratory (see task D4 for an example from bioseparation). The collaboratory is, in essence, a PSE for teaching that will allow students, TAs, instructor(s), and MPSEs to interact for learning about physical systems. In addition, the network approach of our MPSE project is ideal for testing the potential and problems of telecourses. It allows us to integrate geographically disperse experimental, information and software resources in the collaboratory, and use it for distance learning. We are developing two courses using this methodology. A substantial start has already been made in this area, and we plan to offer these courses shortly.

D CONCLUSION

We have attempted to present in this whitepaper an important avenue of research for HPCC, namely the creation of MPSEs and PSEs. We have also attempted to bring out the specific tasks that need to be addressed for such systems to become a reality.

E BIBLIOGRAPHY

1. Richard M. Adler, Emerging standards for component software, *IEEE Computer*, March (1995), 68-77.
2. K.C. Bernard, Ordering chaos: Supercomputing at the edge, in *Technology 2001: The future of Computing and Communications*, D. Leebaert, editor, MIT Press, Cambridge, MA, (1992).
3. J. A. Berninger, R. D. Whitley, X. Zhang and N.-H. L. Wang, A versatile model for simulation of reaction and nonequlibrium dynamics in multicomponent fixed-bed adsorption processes, *Computers in Chemical Engineering*, 15 (1991), 749-768.
4. K. Brockschmidt, *Inside OLE* 2, Microsoft Press, Redmond, Washington (1994).
5. R.E. Burkart, Reducing the R&D cycle time, *Research Tech. Mgmt.*, 37 (1994) , 27-32.
6. 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), 142-152.
7. P.L. Canfield and A. Joshi, A pyramid approach to stereo correspondence, *Proc. First International Conference on Neural, Parallel and Scientific Computations*, Atlanta, 1995.
8. A. Catlin, C. Chui, C. Crabill, E.N. Houstis, S. Markus, J.R. Rice, and S. Weerawarana, PDELab: An object-oriented framework for building problem solving environments for PDE based applications, *2nd Object-Oriented Numerics Conf.*, (A. Vermeulen, ed.), RogueWare Software, Corvallis, OR (1994), 79–92.
9. A.C. Catlin, M.G. Gaitatzes, E.N. Houstis, Z.Ma, S. Markus, J.R. Rice, N.H. Wang, S. Weerawarana, The SoftLab experience: Building virtual laboratories for computational science, *manuscript under preparation*.

10. T.F. Chan, R. Glowinski, J. Periaux and D. Widlund, *Proceedings of the Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, SIAM Pubs (1990).
11. Component Integration Laboratories, *OpenDoc: The New Shape of Software*, Sunnyvale, Calif. (1994).
12. M.A. Cornea-Hasegan, C. Costian, D.C. Marinescu, I. Martin, and J.R. Rice, Towards problem solving environments for high performance computing, *High Performance Computing '94*, National Supercomputer Research Center, Singapore (1994), 354–366.
13. N. Chrisochoides, E.N. Houstis, and J.R. Rice, Mapping algorithms and software environment for data parallel PDE iterative solvers, *Journal of Parallel and Distributed Computing*, 21, 75-95 (1994).
14. T. Drashansky, A. Joshi and J.R. Rice, *SciAgents-* An Agent Based Environment for Distributed, Cooperative Scientific Computing, CSD-TR-95-029, Department of Computer Sciences, Purdue University.
15. T. Drashansky, S. Weerawarana, A. Joshi, R. Weerasinghe, E.N. Houstis, Sofware architecture of ubiquitous scientific computing environments for mobile platforms, CSD-TR-95-032, Department of Computer Sciences, Purdue University.
16. E. Gallopoulos, E.N. Houstis, and J.R. Rice, Computer as thinker/doer: Problem solving environments for computational science. *IEEE Comp. Sci. Engr.*, 1 (1994), 11–23.
17. Michael Gleicher and Andrew Witkin, Through-the-lens camera control, *Computer Graphics*, **26** (1992), 331-340.
18. R. Glowinski, G. Golub, G. Meurant, and J. Periaux, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, SIAM Pubs. (1988).
19. R. Hase, C.G. Takoudis and R. Uzsoy, Cellular and reentrant layouts form wafer fabrication, *Electrochem. Soc.*, **94** (1994), 468-470.
20. E.N. Houstis, J.R. Rice, and S. Weerawarana, A software platform for integrating symbolic computation with a PDE solving environment, *Proc. 14th IMACS World Congress*, IMACS 1, (1994), 482–485.
21. C.M. Hoffmann, E.N. Houstis, J.R. Rice, A.C. Catlin, M. Gaitatzes, S. Weerawarana, N-H L. Wang, C.G. Takoudis, and D.G. Taylor, SoftLab – A virtual laboratory for computational science. *Math Comp. Simulation* **36** (1994), 479–491.
22. E.N. Houstis, J.R. Rice, and R. Vichnevetsky (editors), *Intelligent Mathematical Software Systems*, North Holland, Amsterdam (1990), 363 pages.
23. E.N. Houstis, J.R. Rice, and T.S. Papatheodorou, Parallel ELLPACK: An expert system for parallel processing of partial differential equations. *Math. Comp. Simulation*, **31**, (1989), 497–508. Reprinted in *Intelligent Mathematical Software Systems*, (Houstis, Rice and Vichnevetsky, eds.), North Holland, Amsterdam (1990), 63–73.
24. C.E. Houstis, E.N. Houstis, J.R. Rice, P. Varodoglou, and T.S. Papatheodorou, Athena: A knowledge base system for //ELLPACK. In *Symbolic-Numeric Data Analysis and Learning* (E. Diday and Y. Lechevallier, eds.), Nova Science, New York (1991), 459–467.
25. E.N. Houstis, J.R. Rice, and R. Vichnevetsky (editors), *Expert Systems for Scientific Computing*, North Holland, Amsterdam (1992), 461 pages.
26. E.N. Houstis, J.R. Rice, and R. Vichnevetsky (editors), *Artificial Intelligence, Expert Systems and Symbolic Computing North Holland*, Amsterdam (1992), 458 pages.
27. E.N. Houstis and J.R. Rice, Parallel ELLPACK: A development and problem solving environment for high performance computing machines. In *Programming Environments for High-Level Scientific Problem Solving* (P. Gaffney and E. Houstis, eds.), North-Holland, Amsterdam (1992), 229–241.
28. E.N. Houstis and J.R. Rice, The architecture of PDE solving systems. In *Computer Methods for Partial Differential Equations VII* (R. Vichnevetsky, ed.), IMACS, New Brunswick, NJ (1992), 363–370.
29. E.N. Houstis, J.R. Rice, and S. Weerawarana, A software platform for integrating symbolic computation with a PDE solving environment. *Proc. 14th IMACS World Congress*, IMACS **1**, (1994), 482–485.
30. E.N. Houstis, J.R. Rice, and S. Weerawarana, An open structure for PDE solving systems. *Proc. 14th IMACS World Congress*, 3 (1994), 1296–1299.
31. E.N. Houstis, S. Weerawarana, A. Joshi and J.R. Rice, The PYTHIA project, to be presented at the *First Intl. Conf. on Neural, Parallel and Scientific Computations*.
32. Industrial Research Institute, *Proceedings: Roundtable meeting on reducing R&D cycle time*, Industrial Research Inst., Washington DC, (1992).

33. W.R. Johnson, Jr., Anything, Anytime, Anywhere: The future of networking, in *Technology 2001: The future of Computing and Communications*, D. Leebaert, editor, MIT Press, Cambridge, MA, (1992).
34. Leo Joskowicz, Elisha Sacks, and Vijay Srinivasan, Kinematic tolerance analysis, *Third Symposium on Solid Modeling and Applications*, 1995.
35. Leo Joskowicz and Elisha Sacks, Computational kinematics, *Artificial Intelligence*, **51** (1991), 381-416.
36. A. Joshi, S. Weerawarana and E.N. Houstis, Using neural networks to support intelligent scientific computing, *Proceedings IEEE Intl. Conf. Neural Networks*, Orlando, 1995.
37. A. Joshi, T. Drashansky, E.N. Houstis, S. Weerawarana, *SciencePad*: An intelligent electronic notepad for ubiquitous scientific computing, to be presented at IASTED/ISMM *International Conference on Inteligent Information Management Systems*, June 1995, Washington D.C.
38. A. Joshi, "To Learn or not to Learn ...", *IJCAI workshop on Learning and Adaptation in Multiagent Systems*, Montreal, 1995.
39. A. Joshi, AI, AI, O! , in *AAAI Fall symposium on Improving the Instruction of Introductory AI*, New Orleans, 1994.
40. A. Joshi, T. Drashansky, J.R. Rice, S. Weerawarana, E.N. Houstis, On learning and adaptation in multiagent systems: A scientific computing perspective., *manuscript under preparation*.
41. A. Joshi, S. Weerawarana, E.N. Houstis, J.R. Rice, N. Ramakrishnan, On using computational intelligence to support problem solving environments for scientific computing, *manuscript under preparation*.
42. D.E. Keyes, T. F. Chan, G. Meurant, J. S. Scroggs, and K.G. Voigt, *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, SIAM Pubs. (1992).
43. Sang Bae Kim, A. Hadjidimos, E.N. Houstis, and John R. Rice, The performance of parallel stationary iterative methods for distributed memory machines, *Proceedings of the Intel Supercomputer Users Group*, (1994), 169-173.
44. Sang Bae Kim, S. Markus, ..., Parallel reuse methodologies for "legacy" PDE software
45. J. Ledeberg and K. Uncapher, *Towards a national collaboratory, Report of an Invitational Workshop at The Rockefeller University*, March, 1989.
46. D.C. Marinescu, J.R. Rice, B. Waltsburger, C.E. Houstis, T. Kung, and H. Waldschmidt, Distributed supercomputing. In *Future Trends '90*, IEEE Press (1990), 381–387.
47. E. Mascarenhas, V. Rego, Ariadne: Architecture of a portable threads system supporting mobile processes, CSD-TR-95-017, Department of Computer Sciences, Purdue University.
48. S. McFaddin, *An Object Based Problem Solving Environment for Composite Partial Differential Equations*, Ph.D. thesis, Department of Computer Sciences, Purdue University, 1992.
49. H.S. McFaddin and J.R. Rice, Collaborating PDE solvers. *Applied Numerical Mathematics*, **10**, (1992), 279–295.
50. S. McFaddin and J.R. Rice, RELAX: A platform for software relaxation. In *Expert Systems for Scientific Computing*, (Houstis, Rice, and Vichnevetsky, eds.), North-Holland, Amsterdam (1992), 175–194.
51. M. Mu and J.R. Rice, Modeling with collaborating PDE solvers – Theory and practice. *Contemporary Mathematics*, **180**, (1994), 427–438.
52. Mo Mu and John R. Rice, Collaborating PDE solvers with interface relaxation, *manuscript under preparation*.
53. M. Mu and J.R. Rice, Preconditioning for domain decomposition through functional approximation, *SIAM J. Sci. Comp.*, **15** (1994), 1452-1466.
54. Object Management Group, Common Facilities Architecture Rev. 4.0, *OMG Document No. 95.1.2*, Framingham, Mass. (1995).
55. 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) 554-567.
56. A. Quarteroni, F. Pasquarelli, and A. Valli, Heterogeneous domain decompositions: Principles, algorithms, applications, in *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations (D. Keyes et. al, eds.)*, SIAM Pubs. (1992), 129-150.
57. N. Ramakrishnan, A.Joshi, S. Weerawarana, E.N.Houstis and J.R. Rice, Neuro-fuzzy systems for intelligent scientific computation, CSD-TR-95-026, Department of Computer Sciences, Purdue University.
58. RegoXX
59. J.R. Rice, Learning, teaching, optimization and approximation, in *Expert Systems for Scientific Computing* (Houstis, Rice and Vichnevetsky, eds.), North-Holland, Amsterdam (1992), 89–123.

60. J.R. Rice, The algorithm selection problem. In *Advances in Computers*, Vol. 15, (Rubicoff and Yovits, eds.), Academic Press (1976), 65–118.
61. J.R. Rice and R.F. Boisvert, *Solving Elliptic Problems Using ELLPACK*. Springer-Verlag, New York (1985), 497 pages.
62. J.R. Rice and H.D. Schwetman, Interface issues in a software parts technology. In *Reusability in Programming*, (Biggerstaff, ed.), ITT Technology (1983), 129–137. Reprinted in *Software Reusability* P. Freeman, ed.), IEEE Tutorial, Computer Society Press (1987), 96–104. Revised version in *Software Reusability* (T.J. Biggerstaff and A.J. Perlis, eds.), ACM Press (1989), 125–139.
63. Elisha Sacks and Leo Joskowicz, Automated modeling and kinematic simulation of mechanisms, *Computer-Aided Design*, **25** (1993), pp 106-118.
64. Elisha Sacks and Leo Joskowicz, Computational kinematic analysis of higher pairs with multiple contacts, to appear in *Journal of Mechanical Design*, 1995.
65. C. Taylor and T.G. Hughes, *Finite element programming of the Navier-Stokes equations*, Pineridge Press, Swansea, U.K., (1981).
66. William F. Tidd, James R. Rinderle, and Andrew Witkin, Design refinement via interactive manipulation of design parameters and behaviors, *Proceedings of the 4th Design Theory and Methodology Conference*, 1992.
67. J.T. Vesey, Speed-to-Market distinguishes the new competitors, *Research Tech. Mgmt.*, **34** (1991), 33-38.
68. R. G. Voigt, Requirements for multidisciplinary design of aerospace vehicles on high performance computers, ICASE Report No. 89-70, Sept. 1989, 9 pages.
69. S. Weerawarana, E.N. Houstis, and J.R. Rice, An interactive symbolic-numeric interface to parallel ELLPACK for building general PDE solvers. In *Symbolic and Numerical Computation for Artificial Intelligence*, (Donald, Kapur and Mundy, eds.), Academic Press, (1992), 303–321.
70. S.Weerawarana, *Problem Solving Environments for Partial Differential Equation Based Systems*, Ph.D. Thesis, Department of Computer Sciences, Purdue University, 1994.
71. P. Wu, E.N. Houstis, and J.R. Rice, EPPOD: A parallel problem solving environment for the electronic prototyping of physical objects design, *Proc. DAGS '94 Symposium*, (F. Makedon, ed.), Dartmouth Inst. Adv. Grad. Studies, Dartmouth, NH (1994), 135–151.
72. Poting Wu and Elias N. Houstis, A parallel mesh generation and decomposition methodology, *Proceedings of Mesh Generation Conference*, Albuquerque, Oct. 1994.
73. Science & Technology, *Bussiness week*, May 1, 1995.
74. E. Mascarenhas, F. Knop , V. Rego, ParaSol: A multiThreaded System for Parallel Simulation based on mobile threads, submitted for publication, 1995.
75. Poting Wu, Parallel Shape Optimization, *Proceedings of International Conference on Parallel Algorithms (ICPA'95)*, Wuhan-China, 1995, to appear.
76. Sun Microsystems, The Network is the Computer, Trademark.