The Role of Problem Solving Environments in Watershed Assessment

Clifford A. Shaffer*, Naren Ramakrishnan*, Layne T. Watson*,
Randy Dymond+, and Vinod Lohani#
*Department of Computer Science,
+Department of Civil Engineering, and
#Department of Engineering Fundamentals
Virginia Tech
Blacksburg, VA 24061

August 2, 2006

Abstract

Effective watershed management requires that decision-makers receive input about, and balance consideration of, a number of competing factors. The fundamental drivers of watershed change are modifications to landuse and settlement patterns. To model the effects of landuse and settlement changes properly requires, at a minimum, the ability to model effects related to surface and subsurface hydrology, economics, and biology. This means that an effective decision support system must integrate together several models. At the same time, the users of the system are likely to have diverse backgrounds and levels of expertise, and are certain not to be experts in all of the domains that must be modeled. Problem solving environments (PSE) seek to integrate multiple software tools into a single system for solving decision-making problems. This paper discusses the role that PSEs can and should play in future watershed modeling and management systems.

Keywords: Problem-Solving Environments, Decision Support Systems, Watershed Management Systems.

1 Introduction

Effective watershed management requires that decision-makers receive input about, and balance consideration of, a number of competing factors. The fundamental drivers of change are modifications to landuse and settlement patterns. These changes affect surface and ground waterflows, water quality, wildlife habitat, economic value of the land and infrastructure (directly due to the change itself such as building a housing development, and indirectly due to the effects of the change, such as increased flooding), and cause economic effects on municipalities (taxes raised versus services provided).

To model the effects of landuse and settlement changes properly requires, at a minimum, the ability to model effects related to surface and subsurface hydrology, economics, and biology. This means that an effective decision support system must integrate together several models. At the same time, the users of the system are likely to have diverse backgrounds and levels of expertise, and are certain not to be experts in all of the domains that must be modeled.

Problem solving environments (PSE) seek to integrate multiple software tools into a single system for solving decision-making problems. A PSE should free the user from managing the individual software components. The user should interact with the PSE to specify the parameters of the problem at a high level, rather than in terms of low-level modeling subsystems. The PSE should then integrate the results of the submodels into coherent, visual feedback suitable for high-level understanding.

This paper discusses the role that PSEs can and should play in future watershed modeling and management systems. A companion article [Rubin et al. 2001] describes the current status of a prototype system meant to realize some of the contributions that PSEs can make to watershed assessment. The present paper focuses on the longer-term opportunities that can result from meaningful integration of sophisticated models and effective presentation of simulation results to decision makers.

We will begin with a survey of the current state of the art in decision support systems for watershed assessment on the one hand, and PSEs in general on the other. Next we describe a wishlist of desirable features that should be integrated into a complete PSE for watershed analysis and management. We then address how a number of new software technologies can be used to create such a PSE. Finally, we speculate on what the effects might be once a high-quality PSE for watershed management becomes available.

2 The Current State-of-the-Art

In this section we briefly describe some relevant tools currently available for watershed management, and also describe key features from a few PSEs from other disciplines. This discussion sets the stage for the later sections that discuss what we might hope and expect to see in the future.

Models for key aspects of watershed management have been available for years. An important example is HSPF (Hydrological Simulation Program FORTRAN) [Bicknell et al. 1997], which incorporates a watershed scale agricultural runoff management model and non-point source pollutant loading model into a basin-scale framework. HSPF models hydrological processes mathematically as flows and storages and uses a spatially lumped model for each subarea within a watershed.

HSPF is a commandline-based FORTRAN program that is rather difficult to use even by those familiar with it. The HSPF community has spawned efforts toward creating a PSE with HSPF as the base, by putting a graphical user interface over HSPF. One of these is GenScn (GENeration and analysis of model simulation SceNarios) [Kittle et al. 1998], which attempts to make it easier to enter the necessary data used to drive the HSPF model, and then provide graphs of the results. GenScn is meant the help the user in analyzing various what-if scenarios in a watershed involving land use change, land use management practices, and water management operations. Such scenarios involve analyzing and managing high volumes of input and output data and hence follow a difficult process. GenScn helps in this process by creating simulation scenarios, analyzing results of the scenarios, and comparing scenarios. It provides an interactive framework for analysis built around the HSPF model. The GUI uses standard Windows 9x/NT components. MapObjects LT from Environmental Systems Research Institute, Inc. (ESRI) is used to provide mapping functionality. The model outputs include graphical and tabular display of both observed and simulated data. Output may be viewed interactively or written to files.

PSEs for watershed management are typically centered on physically-based conceptual models which delineate a watershed into multiple classifications based on land use and drainage connectivity. An example is LUCAS (Land Use Change Analysis System) [Berry et al. 1996]. LUCAS is based on a Markov probabilistic model that attempts to capture the influence of market economics (ownership characteristics), transportation networks (access and routing costs), human institutions (population density), and ecological behavior on landscape properties. The primary motivation is socioeconomic modeling. LUCUS uses a Markov transition matrix to assess random spatial variations in landuse which, in turn, are used for assessing the expected impact of a given set of factors.

AQUATOOL, a computer tool for water resources planning and operational management, is composed of modules linked through geographically referenced databases and knowledge bases. These modules are designed to: model water resources schemes optimization, carry out simulation of management of water resources systems including conjunctive use of surface and ground water, and preprocess a groundwater model designed to include distributed aquifer submodels in the simulation model [Andrew et al. 1996]. BASINS, released by the EPA, supports environmental and ecological analysis on a watershed basis through use of models and a GIS [USEPA 1996]. Osmand et al. [1997] developed a DSS called WATERSHEDSS to aid watershed managers in handling water quality problems in agricultural watersheds. The key objectives of this DSS are: to transfer information to watershed managers for making appropriate land management decisions, to assess nonpoint-source pollution in a watershed based on user supplied information and decisions, and to evaluate water quality effects of alternative land treatment scenarios.

Carl et al. [1999] discussed development of a DSS called watershed analysis risk management framework (WARMF) for calculation of the total maximum daily loads (TMDLs) of various pollutants within a river basin. WARMF contains five integrated modules namely, Engineering, TMDL, Consensus, Data, and Knowledge. A GUI that provides menus for the user to issue commands and stores and displays the output in the forms of GIS maps, bar charts, and spreadsheets integrates these modules.

The design of the L2W PSE [Rubin et al. 2001] embodies modeling procedures for the assessment of the hydrologic and economic impacts of alternative landscape scenarios in an integrated framework. Geographic information system (GIS) data and techniques merge both hydrologic and economic models with an intuitive web-based user interface. Using a GIS-based interface produces a more realistic, site-specific application where a user can create a landuse change scenario based on local spatial characteristics. A single interface used to combine the GIS with the physical simulation follows the conceptual model developed by Fedra[Fedra 1993] and Goodchild [Goodchild 1993]. Another advantage of using a GIS with the PSE, as described by [Maidment 1993], is that the GIS can provide necessary parameters to hydrologic and other modeling processes through analysis of terrain, land cover, and other features.

More generally, PSE research includes (1) developing problem-specific PSEs and (2) developing general tools for building PSEs. Issues such as developing a general architecture for PSEs; leveraging the Web; supporting distributed, collaborative problem solving; and providing software infrastructure ("middleware") are also being addressed.

One problem domain where PSEs are common is the numerical solution of partial differential equations (PDEs). An early example is ELLPACK [Boisvert and Rice 1985], a portable FORTRAN 77 system for solving two and three dimensional linear elliptic PDEs. Its strengths include a high-level language which allows users to define problems and solution strategies in a natural way (with little coding), and a relatively open architecture which allows expert users to contribute new problem solving modules. ELLPACK's descendents include Interactive ELLPACK [Dyksen and Ribbens 1987], which adds a graphical user interface to support better user interaction, and Parallel ELLPACK (PELL-

PACK) [Houstis et al. 1998], which includes a more sophisticated and portable user interface, incorporates a wider array of solvers, and can take advantage of multiprocessing. PELLPACK also includes an expert or "recommender" component named PYTHIA [Houstis et al. 2000]. Another system which provides a high level, problem-oriented environment for PDE solving is SciNapse [Akers et al. 1997], a code-generation system that transforms high-level descriptions of PDE problems into customized C or FORTRAN code, in an effort to eliminate the need for programming by hand.

PSEs are being built for a number of other scientific domains as well. For example, Parker et al. [1997] describe SCIRun, a PSE that allows users to interactively compose, execute, and control a large-scale computer simulation by visually "steering" a dataflow network model. SCIRun supports parallel computing and output visualization well, but has no mechanisms for experiment managing and archiving, optimization, real-time collaboration, or modifying the simulation models themselves. Bramley et al. [Bramley et al. 1998, 1998] have developed Linear System Analyzer, a component-based PSE, for manipulating and solving large-scale sparse linear systems of equations. Dabdub and Manohar [1997] have built a PSE for modeling air pollution in urban areas. The WISE environment [Knox et al. 1997] lets researchers link models of ecosystems from various subdisciplines. An object-oriented environment for optimization is DAKOTA [Eldred and Hart 1998], which provides support for legacy code, high level component composition, and parallel computing. Lacking are integrated visualization, collaboration support, experiment management and archiving, and support for modifying the underlying simulation models.

The CACTUS [Allen et al. 1999] system for the relativistic Einstein equations for astrophysics supports distributed computing, visualization, collaboration, experiment management, and model development. Cactus illustrates that, currently, the power and level of integration of a PSE is directly proportional to the specificity of the problems being addressed by the PSE. While CACTUS is quite sophisticated, to adapt CACTUS to a different problem class is likely to be rather difficult, as the component tools are tailored to solving astrophysics equations.

An important goal of PSE researchers is to define a generic architecture for PSEs and to develop middleware (typically object-oriented) to facilitate the construction and tailoring of problem-specific PSEs [Gallopoulos et al. 1994]. This emphasis, along with work in Web-based, distributed, and collaborative PSEs, characterizes much of the current research in PSEs. An example is Catlin et al.'s PDELab [1994], a multilayered, object-oriented framework for creating high-level PSEs. PDELab supports PDESpec, a PDE specification language that allows users to specify a PDE problem in terms of PDE objects and the relationships and interactions between them. Parallel Application WorkSpace (PAWS) [Mniszewski et al. 1998] is a CORBA-based, object-oriented server for connecting parallel programs and objects. Other researchers investigating object-oriented frameworks for PSE building include Gannon et al. [1998], Balay et al. [1998], and Long and Van Straalen [1998].

With the rise of the Web, PSEs are now beginning to support distributed problem solving and collaboration. Regli [Regli 1997] describes Internet-enabled computer-aided design systems for engineering applications. Net PELLPACK [Markus et al. 1997], PELLPACK's Web-based counterpart, lets users solve PDE problems via Java applets. Other Web-based PSEs include NetSolve [Casanova and Dongarra 1997] and NEOS [Czyzyk et al. 1997]. Current PSE-related research projects that emphasize distributed collaboration include LabSpace [Disz et al. 1995], Habanero [Chabert et al. 1998], Tango [Beca et al. 1997], Symphony [Shah and Kafura 1999], and Sieve [Isenhour et al. 1997].

3 What We Would Like to See

There is no doubt that the need exists for better models for all aspects of watershed management, including hydrology (flooding and erosion effects), biology (effects of contaminants and population changes), and economics (valuations resulting from changes in landuse and surrounding environment, economic effects on governments). Yet, models for all of these aspects do exist in one degree or another, and work continues on improving them. Our main concern in this paper is how to leverage these disparate models in ways that best benefit planners and other observers.

The key issues to be addressed by a true PSE for watershed management are integration of disparate models, user-friendly interfaces, and Web-based access. With current advances in network capacity and availability, computing power, and software interfaces, it is now possible to develop systems that will provide better planning capabilities to those who are using the existing tools. Further, with improved access and usability, modern PSEs provide the opportunity to provide advanced planning tools to a much broader audience than ever before.

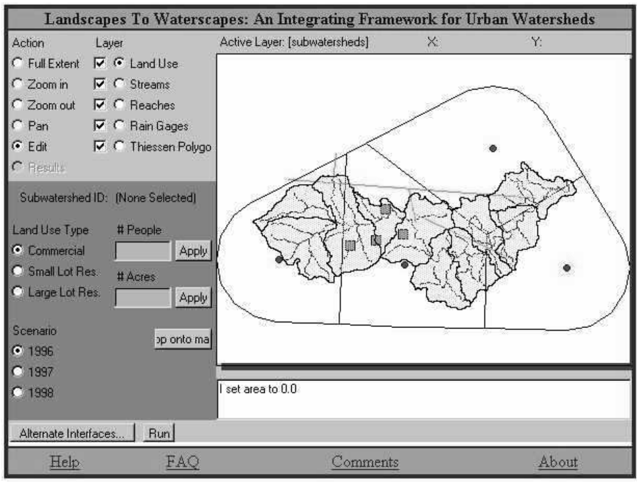
The system we envision would incorporate all of the necessary modeling components to simulate the necessary aspects of a watershed. It would permit the user to set up various initial conditions, being essentially scenario-oriented. That is, the user should be able to see the effect of setting some initial conditions, with the description of those initial conditions being given at a higher cognitive level. For example, current systems such as the HSPF hydrology model require users to enter detailed information about rainfall patterns, geology, and landuse distributions. What would be more useful to a watershed planner would be the ability to model the effects of a major rainstorm after building a large suburban residential community on the side of a particular mountain.

Alternatively, the user should be able to request that the system perform optimization within stated constraints to optimize a user-determined (possibly multi-valued) objective function. The results of running the various simulations should be displayed to the user in a clear way. For example, the planner may know that she wishes to place within a given region a residential community capable of housing 1000 people. Given the various constraints of floodplains, slopes, existing settlement patterns, major roads, etc., it should be possible for an automated optimization routine to select the location that best matches the planners stated objectives.

As an example of how the interface for the PSE might look, consider Figure 1, which shows the primary interface for the L2W watershed management prototype. Interacting with the map, the user chooses a settlement location, the type (one of several choices) and size of the settlement, and a rainfall pattern (from a menu of scenarios), and then invokes a hydrologic simulation. Figure 2 shows the (hydrological) results of one simulation run in L2W. Other outputs include information on economic effects (land values, taxes, infrastructure costs, etc.). While the actual capabilities of L2W are quite crude compared with what we envision here, a more sophisticated system is still likely to be centered upon manipulating a map in some way, and produce the disciplinary (hydrology, economics, biology) outputs in some graphical format.

We now list a number of distinct aspects that should be part of a full PSE for watershed management, along with rationale for the desirability of each point.

Internet Access to Legacy Codes: We have already noted that many of the effects of landuse change can be modeled using existing packages. Rather than create new simulations, it is more effective to integrate existing ones. This is the approach taken, for example, by GenScn, which provides a more effective



Landscapes To Waterscapes: An Integrating Framework for Urban Watersheds

Active Layer: [subwatersheds] X: Y:

Action: ● Edit

Layer: ● Land Use

Subwatershed ID: (None Selected)

Land Use Type: ● Commercial

Scenario: ● 1996

I set area to 0.0

Figure 1: Main interface screen for the L2W system.

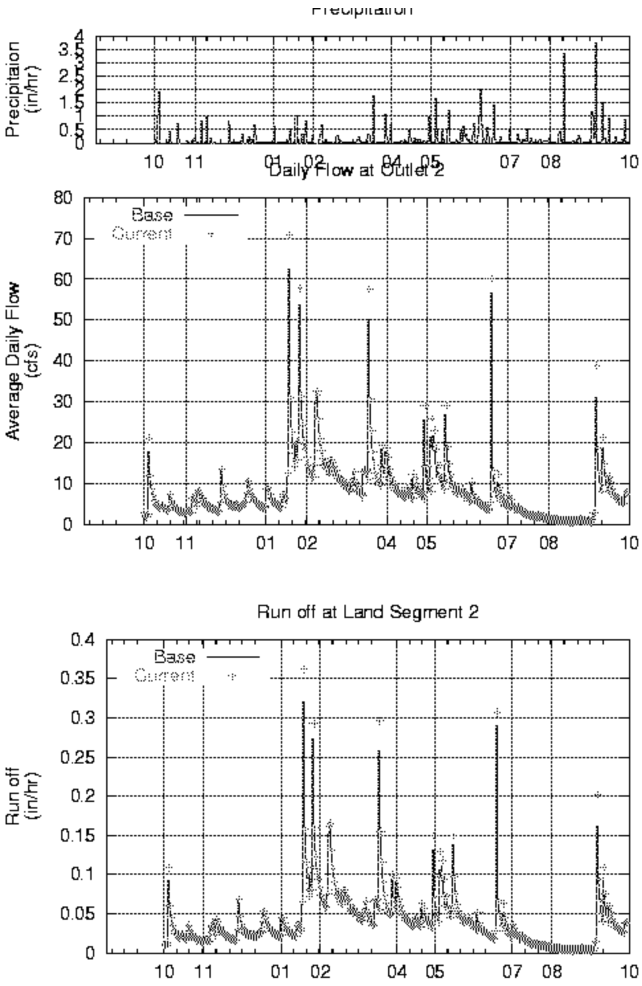


Figure 2: A sample results screen from the L2W system.

user interface to the HSPF hydrology modeling code. The next, crucial step is to link various models together. Of course, this could be done as a stand-alone application on the user's computer. However, important advantages come from making the packages available to users via the Internet. One important advantage is that doing so avoids platform dependency issues. The user is not required to install the system on a compatible platform. Perhaps more importantly, by using a network-based approach, it is not even necessary that all of the models run on the same platform.

Visualization: Users of a decision-making system typically wish to visualize the output, rather than simply analyze the numbers and text produced by the models. The visualization process should be integrated seamlessly with the models. Currently, models such as HSPF will generate a text-based output file. Another program can then be used to read the results and display the information graphically. It becomes a simple matter for a PSE to link together the model that generates the output with the visualization tool.

Scenario Management: Users should be encouraged to experiment with various management options or scenarios. These scenarios should be at a cognitive level relevant to the user, that is, typically higher level than the raw input demanded by the model. As each scenario is evaluated, the results should be storable in a database for later retrieval, and for automated comparison to other scenarios. This is an important issue, especially with the way that many users of models now operate. It is not uncommon for a model user to run the model many times, with various combinations of input parameters, to generate some output that meets some desired criteria. In some cases, users may generate hundreds of experiments. Having a system to automatically track these experiments, even if only to track which parameters generated which output, can be important. If the database of experiments further supports the ability to do "data mining" for desired characteristics, then the users are provided with an analysis tool not currently available to most model users.

Multidisciplinary Support and Usage Documentation: Since the collection of models making up a watershed assessment system are multidisciplinary in nature, the PSE must provide support to users who will not be expert in every (or any) aspect of the problem. This most likely will require alternate interfaces to various aspects of the modeling subsystems to reflect various levels of expertise. This fits well with the notion that scenarios should be at the proper cognitive level for the user. Typically, expert users desire more detailed control of the model. Novice users typically wish to control only the coarse details, and need the maximum amount of guidance on what are reasonable settings for the model. The simulation interface could provide guidance on reasonable interactions of parameters, or which submodels to use in particular circumstances. This guidance regarding parameters must be an integral part of the system for it to be of practical value.

Collaboration Support: Decision makers often would like to either communicate the decision-making rationale to others, or work collaboratively with others during the decision-making process. While the ability to save and restore prior results can be used to provide asynchronous collaboration, ideally a PSE would allow multiple users at multiple workstations to be working together in the PSE at the same time. In this way, one user can create a scenario and display the results to others who are watching. Alternatively, two or more users can jointly set up the scenario.

Optimization: Selecting a "best" configuration to balance the competing goals within a watershed can be cast as an attempt to solve an optimization problem. A given run of the model is typically an evaluation at a single point in a multi-dimensional space. In essence, the goal is to supply to the model that vector of parameters that yields the best result under some figure of merit. As such, the decision-making process can often be improved by applying automated optimization techniques, rather than have someone manually try a large number of parameter sets. Automated optimization techniques are quite sophisticated today, and are woefully underutilized by decision support systems in many disciplines, including watershed management. There exists great opportunity for significant improvement in the value of planning tools with relatively little development effort, since the state of the art in optimization tools far exceeds their current level of use in this domain. As mentioned above, many model users are currently spending large amounts of both human and computer time trying to do what amounts to optimization by hand. That time and computing resource would be better spent with the human acting at a higher cognitive level by describing the evaluation criteria, and then using automated optimization to seek out acceptable solutions that meet those criteria.

High Performance Computing: Many of the models used in watershed assessment require significant computing resources, such as a parallel supercomputer or an "information grid." The PSE should incorporate a computing resource management subsystem such as Globus [Barnard et al. 1999] or Legion, and hide the details of accessing the necessary computation management from the user.

Preservation of Expert Knowledge: Like books in libraries, programs codify and preserve expert knowledge about the application domain. By using and preserving legacy code, the expert knowledge embodied in the legacy codes is (indirectly) employed by the PSE. Yet, state-of-the-art codes in their native from are nearly impossible for nonexperts to use productively. By providing advice (via an expert system shell) the PSE can make the legacy codes and knowledge usable by nonexperts. For decision makers, this expert advice for nonexpert users is indispensable.

Recommender Systems: Ideally, a complete PSE will provide a rich collection of simulations for modeling various aspects of the problem. Unfortunately, the choice available can bewilder novice users. A recommender system for a PSE serves as an intelligent front end and guides the user from a high level description of the problem through every stage of the solution process, providing recommendations at each step.

4 Available Technology

In this section, we describe the present state of various software technologies that should be sufficient to make high-quality watershed assessment systems possible in the near future.

The key need is the ability to link together multiple models, and provide access to the aggregate via the Internet. Fortunately, the techniques for doing this are becoming well understood. "Middleware" refers to software that mediates between a user interface (usually provided via a Web browser) and a back-end database or simulation. Many systems in use everyday by millions of people are based on the middleware model. It is not a difficult matter to write Perl scripts to access the models and visualization tools, and then have a Web server accept commands from the user interface that in turn drive the Perl scripts. Custom Java applets can be used for the front-end interface.

One tool that we have found to be particularly useful for developing a watershed management PSE is MapObjects from ESRI. The purpose of MapObjects is to provide a Web-based interface to ESRI's ARC/INFO product, which is already familiar to many watershed planners. MapObjects provides the ability to call user-defined functions, which in turn permits the ability to access the Perl scripts typically used to drive outside models and visualization tools.

Another alternative is to develop component-based software using, for example, SUN's JavaBeans technology. The goal of JavaBeans is to allow developers to make reusable software components to simplify program development. However, JavaBeans can also be used to develop systems where the "beans" are surrogates for various distributed tools that can be linked together in various ways. Thus, we can envision a system that allows the user to select one or more modeling tools, link them together, and then in turn link the output to the user's choice of visualization tool. Once again, "middleware" software is acting as the intermediary between the various components, taking care of data formating and transfer issues.

The technologies just described for linking together distributed components are now well understood, and currently being used in various PSE systems (see Section 2). Somewhat more speculative is technology for supporting synchronous collaboration. The success of Microsoft's NetMeeting demonstrates that collaborative systems are now reaching the level of limited commercial success. NetMeeting is rather limited in its capabilities, but it is the first practical collaborative system that is widely used by typical users. The research field known as collaborative supported cooperative work (CSCW) is pushing forward on more advanced collaborative systems. Once again, SUN's Java technology provides reasonable possibilities for practical collaborative systems in the near future.

Large-scale simulations can require large amounts of computing power. A plausible alternative to making super-computer class equipment available to local government planners is to harness the computing power that normally goes untapped in desktop computers. A number of efforts are underway to create a computing "power grid." The Information Power Grid [Barnard et al. 1999] being envisioned by NASA and the national laboratories is a general, all-encompassing PSE. While some of the requisite technologies are in place (e.g., Globus [Foster and Kesselman 1997] for distributed resource management, and PETSc [Balay et al. 1997, Gropp and Smith 1994] for a scientific software library), it is unclear how the remaining components can be built and integrated. At this time, IPG is a vision rather than a working prototype.

As the number of algorithms and models made available to the computational scientist increases, there is a concomitant need to support the knowledge-based selection of solution components. This requirement is addressed by recommender systems, introduced in Section 3. Recommender systems are typically designed by organizing a battery of benchmark problems [Ramakrishnan et al. 2000]. and algorithm/model executions, and subsequently mining it to obtain high-level rules that can form the basis of a recommendation.

Such data mining thus constitutes a key computational technology, supporting traditional analysis, visualization, and design tasks [Ramakrishnan and Ribbens 2000] The reader will be familiar with the beers-diapers discovery in commercial market basket analysis ('People who buy diapers in the afternoon are more likely to buy beer too'), but the role of data mining in computational science is a larger and more complicated application. Like most of PSE work, recommender systems research has concentrated on both (i) creating reusable knowledge-bases for specific domains, and (ii) designing software architectures for the rapid prototyping of recommender systems. The PYTHIA kernel, described in [Houstis et al. 2000] provides a database infrastructure for problem and method definition, experiment management, peformance data analysis, and automatic mining of recommendation spaces. Its generic design permits applications to structured domains such as PDEs, numerical quadrature as well as to more amorphous domains, such as watershed management. PYTHIA is built using the Postgres object-relational database system (for storage, retrieval, and management), Tcl/Tk (for interfaces and scripting), statistical software in C (for performance analysis), PROGOL (an induction package for data mining), and CLIPS (a production system shell for making recommendations).

Recommender systems thus contribute directly to automated decision making and also have pedagogical uses in providing phenomenological explanations of their choices and selections. The recently concluded

NSF Sidekic Workshop on PSEs underscores the importance of recommender systems in several key applications [Chandy et al. 1998].

Once recommendations for models are configured, such choices and selections can be optimized to achive user-defined objectives. Multidisciplinary and multiple-objective optimization is a well-understood area of technology, and can thus be deployed immediately in the context of watershed management. There is certainly research work to be done on improved optimization techniques, but standard tools could be integrated with existing models quite quickly.

5 Potential Impacts of Watershed Management PSEs

As should be apparent from the previous section, a number of emerging technologies should soon make possible the creation of more effective watershed management software tools. All of the desired features listed above are within reach. While each item in itself is desirable, the synergy provided by the combination of these features should make the system even more valuable. A collaborative system that provides Internet-based access (perhaps through a Web browser) to an integrated set of models, optimizers, visualizations, and experimental results database, would be a powerful tool indeed.

The most obvious and immediate impact would be on local government planners and resource managers. The ability to predict the environmental consequences of particular development projects, or more generally of policy decisions, is what has been driving advances in GIS and DSS technology for years. What is fundamentally different, and exciting, about the PSEs that should be available in the near future is their ability to be used by a broader audience. Currently, modeling tools such as HSPF can really only be used by those who are already expert in hydrology, and who also are willing to take the time to learn the particular modeling software. This leads to a rather narrow base of direct users.

In practice, the decision makers must rely on technical experts to supply information on which to make decisions, and it is tedious to study various options. In contrast, the PSEs of the future can be made sufficiently easy to use that the decision makers themselves can use the software directly. They will be able to test different scenarios themselves, and directly observe and interpret the modeled results. This could extend beyond government planners to real estate developers, who could make better decisions on which projects to promote.

PSEs can also help to improve education in all of the related disciplines. Students in environmental and civil engineering can more easily be made aware of biological and economic issues, and likewise biologists and economists can acquire some sensitivity regarding issues in the other disciplines. Thus, as these students enter the profession, they will be better able to understand the interrelated nature of the problems they face.

The other group which can hope to benefit from easy-to-use PSEs for watershed management is the general public. Citizens already become involved in controversial zoning and planning decisions. Citizens could hope to go on-line and learn for themselves the various aspects involved in resource management decisions. They could be provided the information to understand the rationale for planning choices made in particular projects, and judge for themselves whether or not reasonable alternatives exist for these choices. Ultimately, a better understanding of the complex issues involved will benefit all parties.

References

[Akers et al. 1997] R. Akers, E. Kant, C.J. Randall, S. Steinberg, and R.L. Young, SciNapse: a problem-solving environment for partial differential equations, *IEEE Computational Science and Engineering 4*, 3(1997), 32–42.

[Allen et al. 1999] G. Allen, T. Goodale, and E. Seidel, The Cactus computational collaboratory: Enabling technologies for relativistic astrophysics, and a toolkit for solving PDEs by communities in science and engineering, in *Proceedings of the 7th Symposium on the Frontiers of Massively Parallel Computation*, IEEE Computer Society, Los Alamitos, CA, 1999, 36–41.

[Andrew et al. 1996] J. Andrew, J. Capilla, and E. Sanchis, AQUATOOL, a Generalized Decision Support System for Water Resources Planning and Operational Management, *Journal of Hydrology 177*, 1996, 269-291.

[Balay et al. 1997] S. Balay, W.D. Gropp, L.C. McInnes, and B.F. Smith, Efficient management of parallelism in object-oriented numerical software libraries, in *Modern Software Tools in Scientific Computing*, E. Arge, A. M. Bruaset, and H. P. Langtangen, Eds., Birkhauser Press, 1997.

[1998] S. Balay, W.D. Gropp, L.C. McInnes, and B. Smith, *A microkernel design for component-based parallel numerical software systems*, Technical Report ANL/MCS-P727-0998, Argonne National Laboratory, 1998.

[Barnard et al. 1999] S. Barnard, R. Biswas, S. Sain, R. Van der Wijngaart, M. Yarrow, L. Zechtzer, I. Foster, and O. Larsson, Large-scale distributed computational fluid dynamics on the information power grid using Globus, in *Proceedings of the 7th Symposium on the Frontiers of Massively Parallel Computation*, IEEE Computer Society, Los Alamitos, CA, 1999, 60–67.

[Beca et al. 1997] L. Beca, G. Cheng, G.C. Fox, T. Jurga, K. Olszewski, M. Podgorny, P. Sokolowski, and K. Walczak, Java enabling collaborative education, health care and computing, *Concurrency Practice and Experience 9*, 1997, 521–534.

[Berry et al. 1996] M.W. Berry, R.O. Flamm, B.C. Hazen, and R.L. MacIntyre, The Land-Use Change Analysis System (LUCAS) for Evaluating Landscape Management Decisions, *IEEE Computational Science and Engineering 3*, 1(??), 1996, 24–25.

[Bicknell et al. 1997] B.R. Bicknell, J.C. Imhoff, J.L. Kittle, A.S. Donigian Jr., and R.C. Johnson, *Hydrological Simulation Program - FORTRAN (HSPF). User's Manual for Version 11.0*, National Exposure Research Laboratory, Research Triangle Park, US EPA, NC 27711, USA, August 1997.

[Boisvert and Rice 1985] R.F. Boisvert and J.R. Rice, *Solving elliptic problems using Ellpack*, Springer-Verlag, New York, 1985.

[Bramley et al. 1998] R. Bramley, D. Gannon, T. Stuckey, J. Villacis, E. Akman, J. Balasubramanian, F. Breg, S. Diwan, and M. Govindaraju, *The linear system analyzer*, Technical Report TR-511, Computer Science Dept, Indiana University, 1998.

[1999] C.W. Carl, J. Herr, L. Ziemelis, R.A. Goldstein, and L. Olmsted, Decision Support System for Total Maximum Daily Load, *Journal of Environmental Engineering 125*, 7(1999), 653-659.

[Casanova and Dongarra 1997] H. Casanova and J. Dongarra, NetSolve: a network server for solving computational science problems *International Journal of Supercomputer Applications and High Performance Computing 11*, 1997, 212–223.

[1994] A.C. Catlin, C. Chui, C. Crabill, E.N. Houstis, S. Markus, J.R. Rice, and S. Weerawana, PDELab: an object-oriented framework for building problem solving environments for PDE based applications, in *Proceedings of the 2nd Object-Oriented Numerics Conference*, A. Vermeulen, ed., Rogue Wave Software, Corvallis, OR, 1994, 79–92.

[Chabert et al. 1998] A. Chabert, E. Grossman, L. Jackson, S. Pietrovicz, and C. Seguin, Java object-sharing in Habanero, *Communications of the ACM 41*, 6(June, 1998), 69-76.

[Chandy et al. 1998] K.M. Chandy, R. Bramley, B.W. Char, and J.V.W. Reynders, *Report of the NSF Workshop on Problem Solving Environments and Scientific IDEs for Knowledge, Information, and Computing (SIDEKIC'98)*, Technical Report, Los Alamos National Laboratory, 1998.

[Czyzyk et al. 1997] J. Czyzyk, J.H. Owen, and S.J. Wright, *NEOS: Optimization on the Internet*, Technical Report OTC-97/04, Argonne National Laboratory, 1997.

[1997] D. Dabdub and R. Manohar, Performance and portability of an air quality model, *Parallel Computing 23*, 14(1997), 2187–2200.

[Disz et al. 1995] T.L. Disz, R. Evard, M.W. Henderson, W. Nickless, R. Olson, M.E. Papka, and R. Stevens, Designing the future of collaborative science: Argonne's Futures Laboratory, *IEEE Parallel & Distributed Technology 3*, 2(1995), 14–21.

[Dyksen and Ribbens 1987] W.R. Dyksen, and C.J. Ribbens, Interactive ELLPACK: an interactive problem solving environment for elliptic partial differential equations, *ACM Transactions on Mathematical Software 13*, 1987, 113–132.

[Eldred and Hart 1998] M.S. Eldred and W.E. Hart, Design and implementation of multilevel parallel optimization on the Intel TeraFLOPS, AIAA Paper AIAA-98-4707, 1998.

[Fedra 1993] K. Fedra, GIS and Environmental Modeling, in *Environmental Modeling with GIS*, M.F. Goodchild, B. Parks, and L. Steyaert, eds., Oxford University Press, 1993, 35–50.

[Foster and Kesselman 1997] I. Foster and C. Kesselman, Globus: A metacomputing infrastructure toolkit, *Internationl Journal of Supercomputer Applications 11*, 2(Summer 1997), 115–128.

[Gallopoulos et al. 1994] E. Gallopoulos, E.N. Houstis, J.R. Rice, Computer as thinker/doer: Problem-solving environments for computational science, *IEEE Computational Science & Engineering 1*, 1994, 11–23.

[1998] D. Gannon, R. Bramley, T. Stuckey, J. Villacis, J. Balasubramanian, E. Akman, F. Breg, S. Diwan, and M. Govindaraju, Component architectures for distributed scientific problem solving, *IEEE Computational Science and Engineering 5*, 2(1998), 50–63.

[Goldin et al. 1998] D.S. Goldin, S.L. Venneri, and A.K. Noor, Beyond incremental change, *IEEE Computer 31*, 10(1998), 31–39.

[Goodchild 1993] M.F. Goodchild, The State of GIS for Environmental Problem-Solving, in *Environmental Modeling with GIS*, M.F. Goodchild, B. Parks, and L. Steyaert, eds., Oxford University Press, 1993, 8–15.

[Gropp and Smith 1994] W.D. Gropp and B.F. Smith, Scalable, extensible, and portable numerical libraries, *Proceedings of Scalable Parallel Libraries Conference*, IEEE, Los Alamitos, CA, 1994, 87–93.

[Houstis et al. 1998] E.N. Houstis, J.R. Rice, S. Weerawarana, A.C. Catlin, P. Papachiou, K.Y. Wang, and M. Gaitatzes, PELLPACK: a problem solving environment for PDE-based applications on multicomputer platforms, *ACM Transactions on Mathematical Software 24*, 1998, 30–73.

[Houstis et al. 2000] E.N. Houstis, J.R. Rice, A.C. Catlin, V.S. Verykios, N. Ramakrishnan, and C.E. Houstis, PYTHIA II: A Knowledge/Database System for Managing Performance Data and Recommending Scientific Software, *ACM Transactions on Mathematical Software 26*, 2(June 2000).

[Isenhour et al. 1997] P.L. Isenhour, J.B. Begole, W.S. Heagy, and C.A. Shaffer, Sieve: A Java-based collaborative visualization environment, in *Late Breaking Hot Topics Proceedings, IEEE Visualization'97*, Phoenix, AZ, October 1997, 13–16.

[Kittle et al. 1998] J.L. Kittle, Jr., A.M. Lumb, P.R. Hummel, P.B. Duda, and M.H. Gray, *A tool for the generation and analysis of model simulation scenarios for watersheds (GenScn)*, U.S. Geological Survey Water-Resources Investigations Report 98-4134, 1998, 152 p.

[Knox et al. 1997] R.G. Knox, V.L. Kalb, E.R. Levine, and D.J. Kendig, A problem-solving workbench for interactive simulation of ecosystems, *IEEE Computational Science and Engineering 4*, 3(1997), 52–60.

[1998] K. Long and B. Van Straalen, PDESolve: an object-oriented PDE analysis environment, in Proceedings of the SIAM Workshop on Object Oriented Methods for Inter-operable Scientific and Engineering Computing, Philadelphia, PA, 1998.

[Maidment 1993] D.R. Maidment, GIS and Hydrological Modeling, in *Environmental Modeling with GIS*, M.F. Goodchild, B. Parks, and L. Steyaert, eds., Oxford University Press, 1993, 147–167.

[Markus et al. 1997] S. Markus, S. Weerawarana, E.N. Houstis, and J.R. Rice, Scientific computing via the Web: the Net Pellpack PSE server, *IEEE Computational Science and Engineering 4*, 3(1997), 43–51.

[Mniszewski et al. 1998] S.M. Mniszewski, P.H. Beckman, P.K. Fasel, and W.F. Humphrey, Efficient coupling of parallel applications using PAWS, in *Proceedings of the Seventh IEEE International Symposium on High Performance Distributed Computing (HPDC7)*, Chicago, IL, 1998.

[1997] S.G. Parker, D.M. Weinstein, and C.R. Johnson, The SCIRun computational steering software system, in *Modern Software Tools in Scientific Computing*, E. Arge, A.M. Bruaset, and H.P. Langtangen (Eds.), Birkhauser Press, 1997, 1–40.

[1997] D.L. Osmand, R. Gannon, J. Gale, D. Line, C. Knott, K. Phillips, M. Turner, M. Foster, D. Lehning, S. Coffey, and J. Spooner, WATERSHEDSS: A Decision Support System for Watershed Scale NonPoint Source Water Quality Problems, *Journal of AWRA 33*, 2(1997), 327-341.

[Ramakrishnan and Ribbens 2000] N. Ramakrishnan and C.J. Ribbens, Mining and Visualizing Recommendation Spaces for Elliptic PDEs with Continuous Attributes, *ACM Transactions on Mathematical Software 26*, 2(June 2000).

[Ramakrishnan et al. 2000] N. Ramakrishnan, J.R. Rice and E.N. Houstis, GAUSS: An Online Algorithm Recommender System for One-Dimensional Numerical Quadrature, *ACM Transactions on Mathematical Software*, to appear.

[Regli 1997] W.C. Regli, Internet-enabled computer-aided design, *IEEE Internet Computing 1*, 1(1997), 39–50.

[Rubin et al. 2001] E.J. Rubin, R. Dietz, J. Chanat, C. Speir, R. Dymond, V. Lohani, D. Kibler, D. Bosch, C.A. Shaffer, N. Ramakrishnan, and L.T. Watson, From Landscapes to Waterscapes: A PSE for Landuse Change Analysis, in *Proceedings of Integrated Decision-Making for Watershed Management Symposium.*

[Shah and Kafura 1999] A. Shah and D. Kafura, Symphony: A Java-based composition and manipulation framework for problem solving environments in *Proceedings of International Symposium on Software Engineering for Parallel and Distributed Systems (PDSE'99)*, May 17–18, 1999, Los Angeles, CA.

[USEPA 1996] USEPA, *Better Assessment Science Integrating Point and NonPoint Sources-User Manual, Version 1.0*, US Environmental Protection Agency, EPA-823-R-96-001, Washington, D.C, 1996.