THE ENVIRONMENTAL DECISION SUPPORT SYSTEM: OVERVIEW AND AIR QUALITY APPLICATION

Steven S. Fine,* John Ambrosiano

Environmental Programs
MCNC–North Carolina Supercomputing Center
Research Triangle Park, North Carolina

1. INTRODUCTION

The Environmental Decision Support System (EDSS) is a computational framework being developed at the MCNC–North Carolina Supercomputing Center. The ultimate aim of EDSS is to help researchers and planners deal successfully with the complexities of the environmental decision-making process. Such factors as:

- estimated risk to health,
- environmental costs,
- regulatory policy and enforcement,
- scientific measurements,
- modeling results,
- social impacts, and
- projected economic costs

are just a few of the many issues that can affect environmental decisions.

As the regulatory process comes under greater scrutiny, the influence of such factors as cost and fairness to affected industries will become even more important. One can expect that the current movement to overhaul environmental regulatory policy will drive the demand for more comprehensive and defensible environmental risk assessment and for routine cost-benefit analyses.

The term "decision support" is sometimes difficult to define precisely. In designing EDSS we have taken an operational point of view. We define decision support as whatever it takes to help people make decisions effectively. From that perspective, what one includes in the design of a decision support system (DSS) depends on the particular problem area. Using this definition, one would design a DSS by studying the decision process in a particular area and looking for opportunities to

- eliminate communication bottlenecks,
- fill in information gaps,
- codify structures and procedures to help manage complexity,
- help decision makers "see" as much of the problem as possible, and
- work to enable balanced, multiattribute decision making.

The first goal of the EDSS project was to help research scientists and planners manage the complexities of making air quality decisions based on model results. Therefore, currently the EDSS framework has several capabilities:

- Model engineering and configuration management services
- Simulation planning and execution management
- Data analysis and visualization
- Experimental strategy development tools

This paper describes EDSS's current capabilities along with example applications. In the future, EDSS will expand to encompass other decision support functions. These will eventually include other kinds of environmental science models, such as surface and groundwater, as well as models for costs and economic effects. EDSS will also incorporate features for managing geographic and other kinds of information. In addition, we envision that EDSS will include a host of decision optimization features.

2. PRESENT EDSS CAPABILITIES

In this section we describe the capabilities that EDSS supplies to help users build air quality models, execute modeling studies, understand the behavior of models and the environment, and develop pollution control strategies. The capabilities are supported on a

*Corresponding author address: Dr. Steven Fine, MCNC, P.O. Box 12889, 3021 Cornwallis Rd., Research Triangle Park, NC, 27709; E-mail: fine@mcnc.org

variety of platforms, including Sun, Silicon Graphics, and Digital Equipment workstations and CRAY supercomputers.

2.1 *Model Structure*

EDSS can manage many types of models, but it has a higher level of support for chemistry-transport models (CTMs) for air quality modeling that follow a modular modeling structure developed by MCNC and the U.S. Environmental Protection Agency (EPA) (Byun et al., 1995). The modeling structure defines calling interfaces for software modules that each simulate a different physical or chemical process, such as gas-phase chemistry, advection, or diffusion. Subroutines that follow the interface standards can be interchanged. For instance, different advection schemes can be tried by simply substituting one for another when the model is built. Modelers who follow the modular modeling structure can easily create and test new algorithms and parameterizations.

2.2 *Model Configuration Management*

A typical modular CTM contains at least seven modules selected from dozens of possible choices, each of which may be composed of several FORTRAN files. EDSS contains tools that assist in the managing module source code and building models. The module source management tool tracks changes to each module's source code and provides convenient access to different versions of the code. For instance, the tool allows a group of modelers to share ongoing changes to an advection module while other modelers access an older but more stable version of the code. The source management tool provides services to local and remote computers, which prevents situations where versions of code on different computers diverge. The code manager's graphical user interface displays the status of source files and provides a convenient way to register new changes. We have found that the source management tool has allowed modelers to improve and share modules without losing code or interfering with each other's work.

EDSS's model building tool makes it easy to reliably construct modular models. This facility has some similarities to the UNIX *make* utility, but the EDSS facility directly supports the concept that models or programs are built from modules that may have different versions and may reside on another computer. To build a model, a user simply selects which modules should be included and optionally which versions of those modules should be used. EDSS determines which files and versions are required, retrieves those versions if necessary, determines which compilations are necessary, and builds the model. The model builder has improved the reliability and reduced the difficulty of creating complex models from research codes.

Figure 1 is an example of a configuration file, which describes the components of a model and how the model should be constructed. Each "module" statement specifies a module that should be included. The first time the model builder is applied to this configuration file, the builder will retrieve the dozens of source files contained by the modules, possibly retrieving the files from a remote computer. The model builder will then compile and link the files. If the same configuration file is used later, the model builder will automatically retrieve and compile only those files for which new versions are available.

The EDSS source management and model building facilities have been applied to other types of programs beyond modular models. For instance, the source management tools are used to manage the source code for EDSS itself. The tools also manage update decks and scripts that are required to use the Penn State/National Center for Atmospheric Research Mesoscale Modeling System (MM5) (Grell et al., 1995). We do this to track our own modifications to MM5 and to keep code consistent among multiple computer systems.

Future versions of EDSS will provide expanded model management capabilities. This will include restricting which sets of modules can be combined (for instance, an aqueous chemistry module might require that a particular cloud scheme be used) and providing additional automation for specifying grids and for building nested models.

2.3 *Computational Study Planning and Execution*

Another problem that can be even more complex than building models with the correct code is to correctly execute a set of related programs. For instance, a typical model run might require executing a dozen preprocessors in the proper order, executing the model, and then running several analysis programs. In the past, mistakes in managing which data sets and programs are used have caused serious problems, some of which have not be recognized until long after the mistakes have been made.

To address this problem, EDSS allows a user to plan and execute computational "studies," where a study is a set of programs, related data sets, and additional information about the computations to be performed. A user builds a study with a mouse, creating icons that represent programs to be executed and creating links between the icons to represent data dependencies between the programs (Fine et al., 1994). For instance, Figure 2 shows part of a study that transforms inventories of atmospheric emissions into a form that is suitable for input to a CTM. The rectangles in the right part of the study represent programs that transform the data. Studies can contain many types of programs. Studies have executed complex “legacy” regulatory systems, such as the one in Figure 2, and our own advanced research models with experimental visualization tools. For each program in a study, the user can select which files should be consumed and produced. The lines between the boxes represent the flow of data between programs.

The information in the left portion of the study helps determine the behavior of the study. For instance, configuration files that are used by many programs can be set. During the planning process, a user can select which computer should be used to execute a study or an individual program in the study. Creating even a simple study can eliminate the requirement for hundreds of lines of shell scripts.

After a user has planned a study, he can execute it. EDSS verifies that all required files are available and executes programs on the specified machines. EDSS graphically indicates which programs are running and which have completed successfully or have failed. After the execution of the study has completed, EDSS maintains a record of the computations that were performed in case questions arise later.

We plan many enhancements for EDSS study management capabilities. Greater assistance will be provided in constructing studies. For instance, EDSS will contain enough information about program capabilities to provide suggestions in some cases for how a study should be built to achieve specified goals. To support more complex sets of programs, EDSS will allow loops and branches within studies and will allow studies to be nested within other studies.

2.4 *Data Visualization*

Data visualization within EDSS is provided by a subsystem called the Package for Analysis and Visualization of Environmental data (PAVE). PAVE is part of EDSS, but it can also be used independently of the rest of EDSS.

PAVE can be used to efficiently and flexibly analyze large gridded data sets, which are typical in air quality applications. PAVE reads ASCII, Urban Airshed Model, and EDSS/Models-3 file formats. Files can reside locally or on remote systems. Subsets of data sets can be extracted based on time period, vertical layer, or arbitrary selections of grid cells. When accessing data sets on remote machines, PAVE transfers only the minimum amount of information between machines, which significantly improves performance and reduces network loads. For instance, if a user runs PAVE on a workstation and requests to view the bottom layer of a multilayer data set that resides on a CRAY, a PAVE process running on the CRAY will extract the data for the bottom layer and send them to PAVE on the workstation.

PAVE provides algebraic and statistical operations that can be applied to data sets. Typical calculations are to sum the concentrations of related chemical species, to compute percent differences in results from two model runs, and to determine domain-wide averages. Commonly used formulas can be saved and applied repeatedly.

The values that result from data extraction and manipulation can be viewed in a variety of ways. Chemical concentrations are frequently displayed in a “tile plot” that colors each cell in a layer to indicate the concentration in that cell. PAVE allows a user to enlarge areas of tile plots, to see the numeric values in selected regions of a plot, and to synchronously animate multiple plots. PAVE can display the magnitude of values in a layer with a pseudo-3D surface plot. PAVE also creates time series and scatter plots that illustrate the relationship between two variables.

PAVE has been designed to require very modest amounts of memory and processing time. This allows analyses to proceed quickly on most workstations. We will extend PAVE in the future to include additional data manipulation and visualization capabilities.

2.5 *Strategy Development*

An important goal of many environmental quality managers and analysts is to develop and understand strategies for reducing environmental side effects of human activities. For instance, many states have developed approaches to reduce ozone levels in their jurisdictions. Options that might be considered for reducing ozone include limiting power plant emissions, requiring the use of reformulated gasoline, implementing stricter vehicle inspection programs, and restricting the placement of new industries. Each combination of options has benefits and costs that must be considered. The EDSS capabilities that are described above reduce the effort required to evaluate alternatives. Several other EDSS capabilities are under development that will further assist in the efficient consideration of alternatives.

One of the bottlenecks to modeling the air quality effects of a control strategy has been the computer time required to transform emissions information for thousands of sources into a form that can be used in the CTM. Typically, changing emissions information for one source, such as adding a simulated scrubber to a plant, has required reexecuting a significant number of computations. To address this problem, EDSS incorporates a new concept in emissions processing that significantly reduces the amount of recomputation that is required when a new control strategy is evaluated (Coats, 1996). The EDSS emissions processor can reduce the computation time required for processing emissions information from hours to minutes. That improvement can significantly reduce the time required to evaluate a potential control strategy.

EDSS also incorporates an experimental capability that allows a user to easily view, in maps and tables, major point sources in an emissions inventory, such as factories. The user can easily access characteristics of each source, such as the mass of pollutant emitted and which pollution control technologies can be applied. A user can easily select a control technology, such as a scrubber, for a source and see estimates of the reduction in emissions and the additional monetary costs that would be incurred if the control were applied. This experimental capability allows users to easily examine contributors to an air quality problem and iteratively explore possible solutions.

Another experimental capability in EDSS is the application of optimization techniques to the development of control strategies. EDSS can find the least costly method of obtaining a specified reduction in pollutants. This is useful when sensitivity analyses for a domain show that an *X*% reduction in pollutants might produce the desired results. EDSS can also use an air quality model in conjunction with the optimization to determine the least costly way of reaching a specified level of air quality (Loughlin and Ranjithan, 1995). These optimization techniques can provide ideas for comprehensive control strategies that might not normally have been considered.

3. CONCLUSION

EDSS provides several capabilities that assist scientists and environmental managers in air quality modeling and decision making. EDSS makes modeling easier and less error-prone, which allows users to better concentrate on environmental issues. We are building on the existing capabilities to provide increasingly advanced features to assist users. This will include providing suggestions for policies that might not normally be considered and providing additional information, such as costs of approaches, that has not been readily available. We will also be extending EDSS beyond air quality to help improve the effectiveness and efficiency of decision making for other environmental issues.

Current information on EDSS and related work is available at “http://www.iceis.mcnc.org”.

Acknowledgments: This work is being supported by EPA cooperative agreement CR-822066.

REFERENCES

Byun, D. W., C. J. Coats, D. Hwang, S. Fine, T. Odman, A. Hanna, K. J. Galluppi, 1995: Prototyping and implementation of multiscale air quality models for high performance computing. *Proceedings, SCS 1995 Simulation Multi-Conference*, Phoenix, AZ, Soc. for Computer Simulation, 527-532.

Coats, C. J., Jr., 1996: High performance algorithms in the Sparse Matrix Operator Kernel Emissions (SMOKE) modeling system. Accepted for *Proceedings of the 9th Joint AMS-A&WMA Conference on the Applications of Air Pollution Meteorology*, Atlanta, GA, Amer. Meteor. Society.

Fine, S. S., S. Chall, D. Hwang, 1994: A human-computer interface for managing complex computational tasks. In *Preprints, 10th International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology*, Nashville, TN, Amer. Meteor. Soc., 32-36.

Grell, G. A., J. Dudhia, D. R. Stauffer, 1995: A description of the fifth-generation Penn State/ NCAR Mesoscale Model (MM5), NCAR Tech. Note NCAR/TN-398+STR, 122 pp.

Loughlin, D. H., and S. Ranjithan, 1995: An application of genetic algorithms in air quality management. *Proceedings of the Second Congress on Computing in Civil Engineering*, Atlanta, GA, Amer. Soc. of Civil Engineers, 963-970.

```
//
// This is a configuration file for one instance of a CTM
model ctm;

// Include files that are used by multiple modules
include SUBST_HGRD_ID /pub/storage/edss/aqm/src/icl/hgrid/HGRD3.EXT;
include SUBST_VGRD_ID /pub/storage/edss/aqm/src/icl/vgrid/VGRD3.EXT;
include SUBST_SPCS_ID /pub/storage/edss/aqm/src/icl/species/SPCS3_RADM.EXT;
include SUBST_CONST_ID /pub/storage/edss/aqm/src/icl/shared/CONST3_RADM.EXT;
include SUBST_PARMS_ID /pub/storage/edss/aqm/src/icl/shared/PARMS3.EXT;
include SUBST_FDESC_ID /pub/storage/edss/aqm/src/icl/shared/FDESC3.EXT;
include SUBST_IODECL_ID /pub/storage/edss/aqm/src/icl/shared/IODECL3.EXT;
include SUBST_FILES_ID /pub/storage/edss/aqm/src/icl/shared/FILES3_RADM.EXT;
include SUBST_ADV3_ID /pub/storage/edss/aqm/src/icl/shared /ADV3_RADM.EXT;

module radm_smo development;  // advection -- development version
   f77_flags "-g";      // compile this module with debugging information
   hadvec.F // This file is being edited locally.
            // Use the local, not the archived version.

module radm2_chem;      // chemistry

module radm_eddy; // diffusion

// cloud module
// cldnoop is a "null" routine.  Using this module turns off cloud
// calculations, which can be useful for evaluating the other modules.
module cldnoop;

module couple;
module radm_driver;
module init;
module addrate;
module getstep;
```

Figure 1: A configuration file that describes the components of a modular chemistry-transport model. Text following "//" is a comment. This configuration file provides enough information for the model to be built.