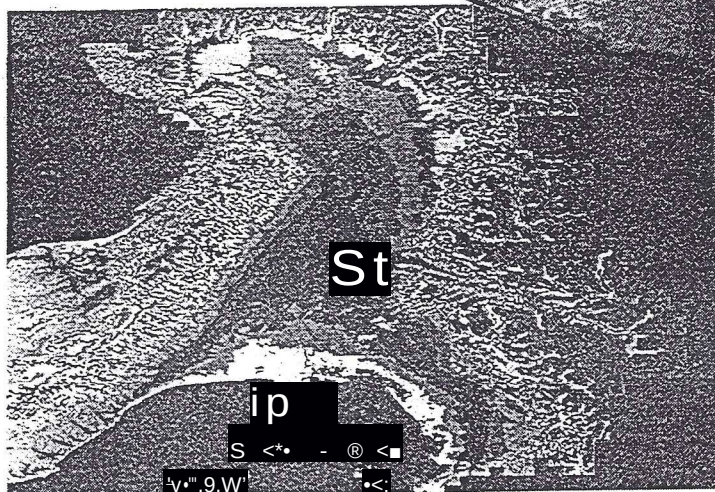
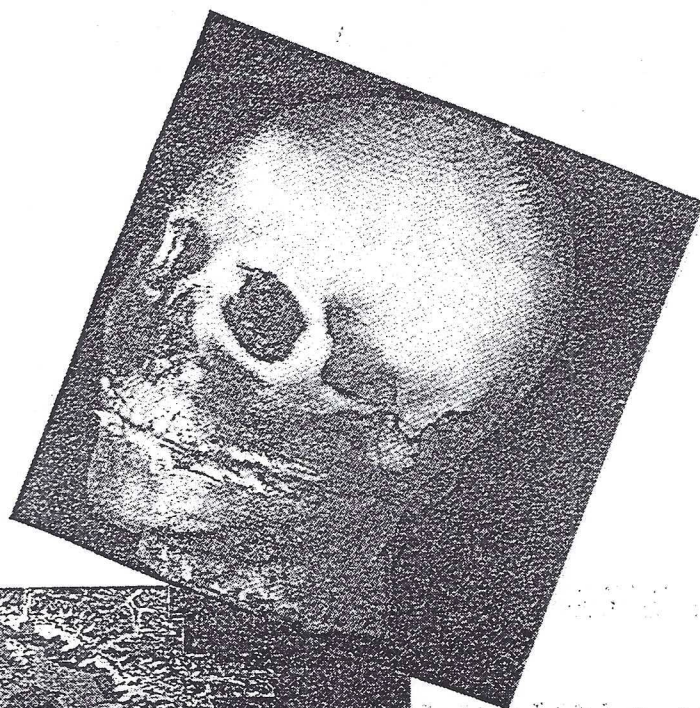
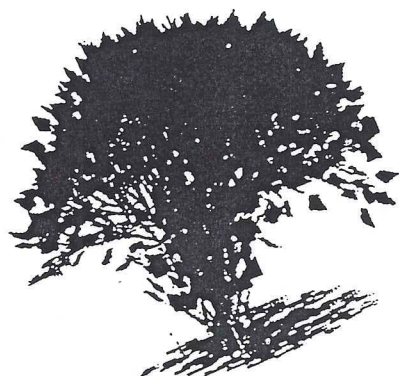




British Computer Society *Displays Group*



International Conference *Visualization and Modelling*

University of Leeds
6 and 7 December 1995

An Integrated Visualization System For Modelling Pollutant Migration

M.D. Impey, K.J. ClarlJ, J.P. Humm[†],
S. ShalP, M.J. Williams and I. Curington*

†Inters Information Technologies Ltd
Environmental Division
Chiitem House
45 Station Road
Henley-on-Thames
Oxon RG9 1AT, UK

*AVS/UNIRAS Ltd
Montrose House
Chertsey Boulevard
Hanworth Lane
Chertsey
Surrey KT16 9JX, UK.

Abstract

We discuss the growing demand for quantitative assessment of the impact of pollutants in the environment, and the need to communicate results of assessment studies to non-specialists and decision makers. We argue that integrated/interactive modelling systems have the potential to meet these demands particularly if visualization tools are used to provide users with direct access to input and output data. In this paper we focus on the issues that need to be tackled in designing an integrated and interactive system for modelling subsurface pollutant migration, and describe the design principles which have been adopted in the development of one such application, MACRO-AFFINITY.

1 Introduction

There is a growing demand from consumers and legislators that the environmental impact of industrial processes is restricted to acceptable levels. To meet this demand often requires a detailed quantitative assessment of the fate of pollutants released into the environment. For processes such as the underground disposal of waste (radioactive; toxic; mixed-waste) a key factor in a quantitative assessment is understanding the flow of groundwater through subsurface rocks and investigating how pollutants migrate through the groundwater and this effect on groundwater quality.

As a result mathematical and computer codes for models of groundwater flow and pollutant migration have been extensively developed and are widely available. A major problem now, is not computing a solution, but instead interpreting and communicating the significance of the computed solutions to non-specialists and decision-makers. A key cause of this problem is that running models on realistic scenarios often requires a daunting range of specific mathematical, numerical and computational skills. Problem specification is often

undertaken by editing a text input file, and viewing of results is done by porting results to a separate graphics viewer. This has resulted in many of the codes being used in batch mode, with a mathematical specialist running the code, a decision-maker reviewing the output results and returning requests for changes in problem specification, input data or output data format to the mathematical specialist. The inflexibility of this system precludes decision-makers from concluding a series of rapid what-if calculations and restricts the number of times the process can be applied iteratively to reach a solution.

What is required is a self-contained and interactive system enabling a non-mathematical expert to specify and solve a problem and have full access to the computed results. We argue that full interaction is best achieved by using visualization, to collect information from a user, provide dynamic feedback and display computed results. This poses a considerable challenge since the visualization aspects of the system should neither dominate the system to such an extent that basic functionality is compromised nor present the user with information in a potentially misleading manner.

Here we discuss the experience gained in developing a fully integrated, interactive visualization system for subsurface pollutant migration. The focus is on the design of such a system, called MACRO-AFFINITY that was developed by Intera using the visualization/development tools available in the software package AVS (Version 5.0) [1]. The paper continues, in Section 2, with a detailed description of the type of problems that are tackled in modelling subsurface pollutant migration and the key features that need to be specified in setting up a model solution. The following section (Section 3) sets out the principles used in designing an integrated, interactive visualization system to solve such problems. The implementation of the design principles is illustrated by a description of the MACRO-AFFINITY modelling system. Particular emphasis is placed on what information needs to be provided by a user at different stages in solving a problem, and what and how information is to be visualized. Section 4 describes the AVS software used in the development of MACRO-AFFINITY focusing on the efficient implementation of design. Section 5 contains concluding remarks and comments.

2 Groundwater Pollutant Problems

The specification and solution of a groundwater pollutant problem can, typically, be split into the following stages:

1. specification of the model region physical properties;
2. specification of the flow boundary conditions;
3. solution of the steady-state flow equations;

4. specification of the pollutant sources;
5. solution of the pollutant migration equations.

In the following sections we consider each of these stages in more detail, with particular attention to the data required to undertake each step.

2.1 Specification of Model Region

In specifying the model region, basic geometric details are needed first. Although some calculations are done in three dimensions the current preference is for **two-dimensional** calculations in a region of a given **thickness**. The main reason for this is that often migration of a pollutant in a particular geological strata is of interest. Therefore specification of maximum and minimum (x, y) values and a thickness (in the direction perpendicular to the (x, y) plane) are sufficient.

The focus of many studies is to consider how details of the shape and size of a repository or its position relative to particular geological features (e.g. impermeable strata; fracture zones; faults) affects the migration of pollutants from the repository. It is thus important to be able to specify sub-regions in the model region, with each sub-region representing a different natural or man-made material.

Within each sub-region two specific flow properties have to be set:

- A porosity;
- Zf permeability.

The first provides a measure of how much of a material consists of open pores or fractures which can be accessed by water. The second indicates the ease with which water can pass through a material under a given forcing pressure gradient. A series of studies have shown that both the porosity and permeability of a rock are complex spatially varying functions, but that it is variations in the permeability that have most effect on pollutant migration. Thus the porosity is often regarded as a constant and attention is focused on characterizing and representing the **spatial variability** of the permeability. The main reason for this is that small length scale variations in the permeability can affect pollutant migration behaviour over much larger length-scales. Thus obtaining a detailed high resolution map of the permeability is a key step in realistic modelling of pollutant migration.

Permeability measurements are often available at a number of points in the model region. A high resolution **permeability map** can be obtained from

this data using geostatistical methods first to quantify the spatial variability of field/experimental measurements of permeability, and secondly to generate a map (on a specified high-resolution grid) that interpolates the measurements and has the same mean, variance and spatial variability properties as the measurements.

2.2 Specification of Flow Boundary Conditions

Having specified the geometry and physical parameters of the model region the next objective is to determine how groundwater flows through it. It is usually Assumed that any migrating pollutants will be transported passively by the groundwater without affecting the groundwater flow and that density and thermal effects are negligible. Thus the solving for the groundwater flow can be conducted independently from and prior to solving for pollutant migration. Here we also restrict attention to migration through regions fully saturated with groundwater.

The groundwater is assumed to satisfy the following standard equations:

$$u = -\frac{k}{\mu} \nabla P \quad (2.1)$$

$$\nabla \cdot u = 0 \quad (2.2)$$

where u is the groundwater velocity, k is the permeability, P is the water pressure, and μ is the water viscosity. Notice that these equations are time-independent so that once the permeability is specified the only additional information required is that the **pressure** is set at two or more points in the model region. It is usual for the pressure to be set to a specified value along an external boundary of the model region. An additional constraint that is sometimes applied is that the flux across a given segment of the model region's external boundary is zero. Although these boundary conditions seem relatively simple, this partly because the model region is often chosen with the specific intention that simple flow boundary conditions are applicable.

2.3 Solution of the Groundwater Flow Equations

Provided valid pressure boundary conditions have been set, the solution of the groundwater flow equations is a relatively straightforward numerical exercise. The main action required is the specification of a **numerical grid** (finite difference or finite element) over which the governing partial differential equations (2.1), (2.2) are discretized. It is standard practice to use either the same grid as used in generating the permeability map, or a subdivision of it. Most numerical methods will also require the setting of a parameter against which the numerical convergence of the flow solution can be tested.

2.4 Specification of Pollutant Sources

The starting point for the specification and solution of the pollutant migration equations is to define where and how pollutant enters the model region. Since the **source of pollutant** being considered is usually a repository or other concentration of pollutant, a point source or sub-region source is most often used. The temporal behaviour of the source can be very complex but, most often, sufficient information can be obtained by considering either a pulse pollutant input or a constant decaying source.

2.5 Solution of the Pollutant Migration Equations

A pollutant migrating through a water-saturated rock may be subject to a number of physical and chemical processes: molecular diffusion; adsorption to the rock; chemical reaction with other species in the groundwater; radioactive decay; and chemical degradation are most commonly considered. The numerical simulation of the migration of a chemically active pollutant is a very challenging task that is still the subject of much current research [2] so attention is usually confined to modelling the migration of chemically inert pollutants.

Two numerical approaches to the numerical simulation of pollutant migration are currently in use. The first is a **continuum model**, based on a system of partial differential equations describing the spatial and temporal variation of pollutant concentration. The second is a **particle tracking** approach in which the movement of discrete packets of pollutant is tracked through the model region. Both approaches have been the subject of detailed research and quick, robust algorithms are available [3], [4]. The algorithms generally require three forms of input. First, **parameters** for each of the processes the pollutant will be subject to (for example, the diffusion coefficient, or the radioactive decay rate). Secondly, a **numerical grid** on which the continuum partial differential equations are discretized or which forms the spatial discretization of the model region for particle tracking algorithms. It is standard practice to use the same numerical grid for solving the pollutant migration equations as used in solving the groundwater flow equations (see Section 2.3). The third input is a **time-step** for the algorithm, which is required because the position of the pollutant is time-dependent and so the numerical algorithms utilize a temporal discretization.

2.6 Analysis of Results

The above subsections indicate the basic **input** required to specify and allow numerical solution of groundwater pollutant migration problems. Before considering the design of a visualization system for such problems, we need to consider what **output** (visual or otherwise) will be required at each stage of the

specification and numerical solution process.

2.6.1 *Output from specification of model region*

Given the discussion in Section 2.1, there are two main outputs from the specification of the model region. First, the division of the model region into **sub-regions** and secondly the **permeability map**. An additional derived output is the numerical grid on which the permeability map is defined.

2.6.2 *Output from the groundwater flow equations*

The two main outputs from the numerical solution of the groundwater flow equations are the computed **steady-state velocity** and **pressure** values at each numerical grid point in the model region. A third output is the numerical grid on which the velocity and pressure were computed.

2.6.3 *Output from the pollutant migration equations*

The pollutant migration equations are different from the groundwater flow equations in that their solution is time-dependent. Three outputs are of fundamental importance in interpreting pollutant migration. First, the temporal evolution of the **plume** as pollutants migrate from the source. Secondly, the temporal variation of the concentration at particular points. The results at these **monitor points** provide quantitative checks and are useful in cross-checking the numerical results with field or experimental measurements of pollutant concentrations. Thirdly **breakthrough curves** which record the amount of pollutant crossing the external boundaries of the model region. These outputs indicate to an engineer/decision-maker the time and location at which pollutants are predicted to enter potentially sensitive areas (for example, underground drinking water supplies) and the amount of pollutant doing so.

3 An Integrated and Interactive Modelling System

3.1 Design Principles

The input and output requirements listed above are both well known and widely accepted, and form the basis for a number of computer codes for solving groundwater pollutant migration problems. Given the discussion in Section 1, we now consider how these basic requirements can be extended to form the design basis for a modelling system integrating problem specification, problem solution and results analysis, allowing interactive modelling, rapid prototyping and what-if

calculations. The top-level design principle is to exploit graphical user interface (GUI) and visualization tools to allow the user to undertake steps in problem specification, solution and diagnostics by direct graphical interaction with output results from previous steps.

A number of implementation-level design principles have been adopted by Intera in the development of integrated, interactive modelling systems:

- A** divide problem into identifiable and manageable sections;
- A** adopt a modular approach;
- A** utilize third-party GUI and visualization software;
- A** embed numerical solvers in modules, maintaining independence between numerical and visualization data structures;
- A** allow multi-level input and output;
- A** modules are responsible for checking if they have sufficient and/or correct input data to run;
- A** modules are responsible for local data structures.

The best way to describe the use of these principles is by illustration of their application. In the following subsections we describe the design of one modelling system, MACRO-AFFINITY, which used AVS (Version 5.0) as a development framework and provider of GUI and visualization tools.

3.2 Data Flow and Control

The sequential nature of the problem specification and solution process, described in Section 2, means that the modelling system can be divided into four distinct sections each corresponding to a group of modules:

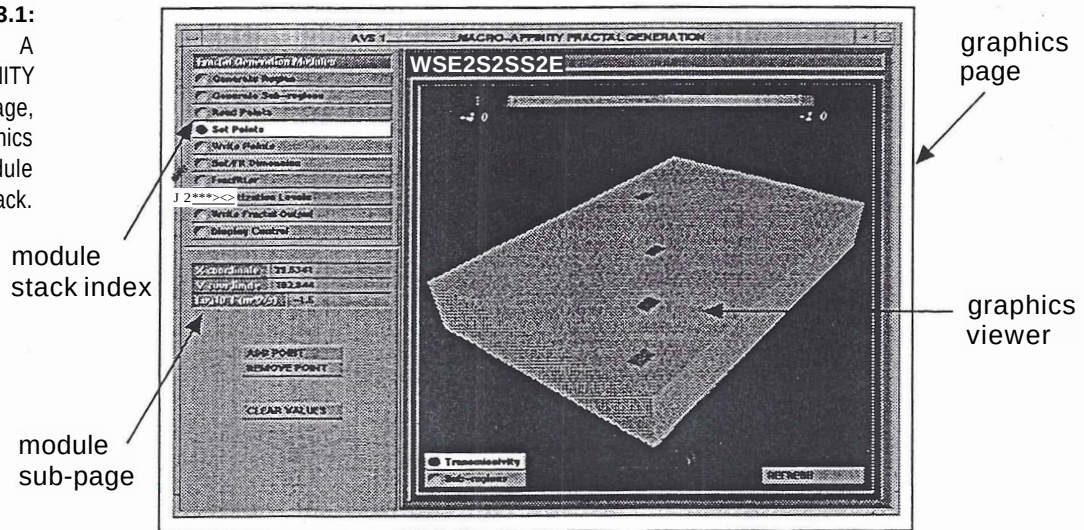
1. specification of model region;
2. specification and solution of groundwater flow equations;
3. specification and solution of pollutant migration equations;
4. review of pollutant migration results.

The GUI for MACRO-AFFINITY consists of four main pages (referred to as Page 1,2,3, and 4 in the following discussion), corresponding to the four sections. An additional page consisting of a "Quit" button and four toggle

(on/off) widgets, showing or hiding each of the four pages provides top-level control.

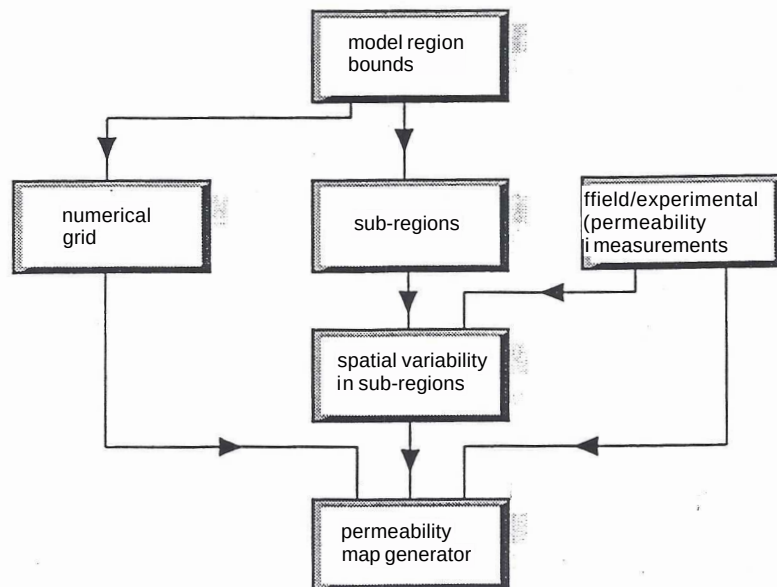
Each of the four main pages consists of a graphics viewer and a module stack (Figure 3.1). Each module allows the user to carry out a specific task (for example: define (x, y) bounds for model region; draw sub-regions; etc.).

Figure 3.1:
A
MACRO-AFFINITY
graphics page,
including graphics
viewer and module
stack.



Module widgets (such as toggle buttons, dials, type-in boxes) which allow the user to control the execution of the module are placed on a module sub-page. Rather than display all the module sub-pages at once, only one sub-page is visible at a time. The module sub-page is displayed immediately adjacent to a stack index, which consists of a column of toggle buttons each labelled with a module name. On clicking a stack index button, the corresponding module sub-page appears. The stack index and the module sub-pages form the module stack.

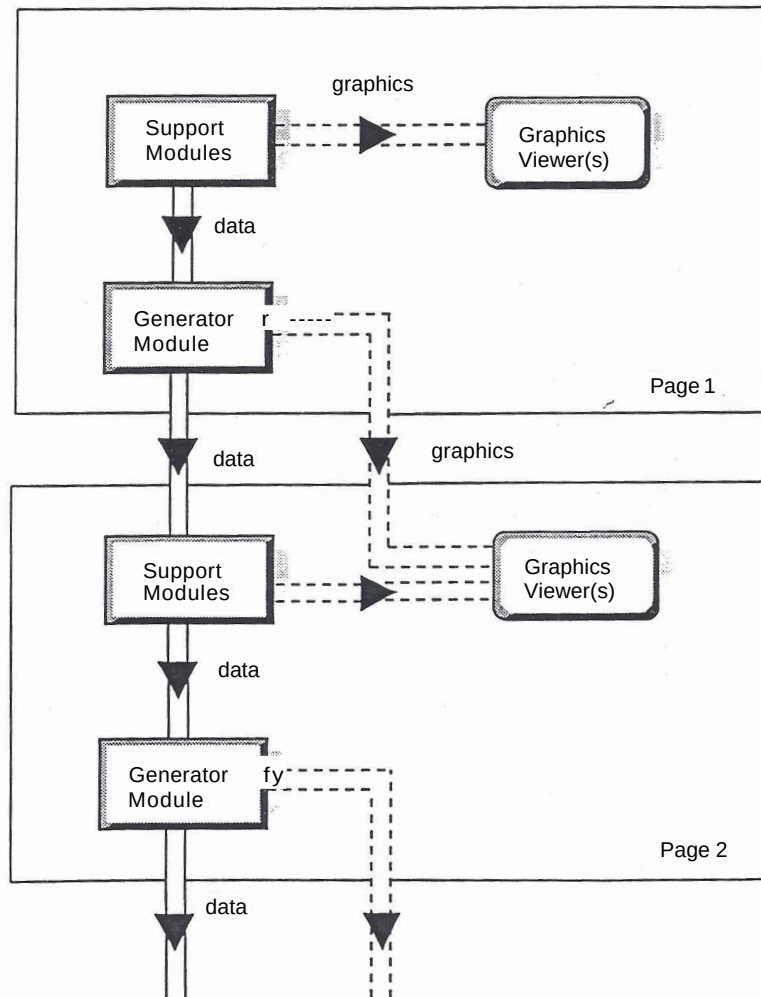
Figure 3.2:
Schematic of the
support module
hierarchy for Page
1.



In the group of modules associated with a page, one module is thought of as being the key output generator: the **generator module**. For example, when specifying the model region the key output is the permeability map. The additional **support modules** each provide independent input information to this output generator in a hierarchical manner (Figure 3.2).

Individual support modules will only execute if data is available from other support modules prior to them in the hierarchy. Data from the support modules is used to update a graphical representation of the data to be input to the generator module. This graphical representation appears in the graphics viewer (or a supplementary graphics viewer) on the page on which the support modules are located. Data from the generator module is used as input to the support modules and graphics viewer on the next page (Figure 3.3).

Figure 3.3:
Schematic showing
the
support-generator
design and the link
between
successive pages.

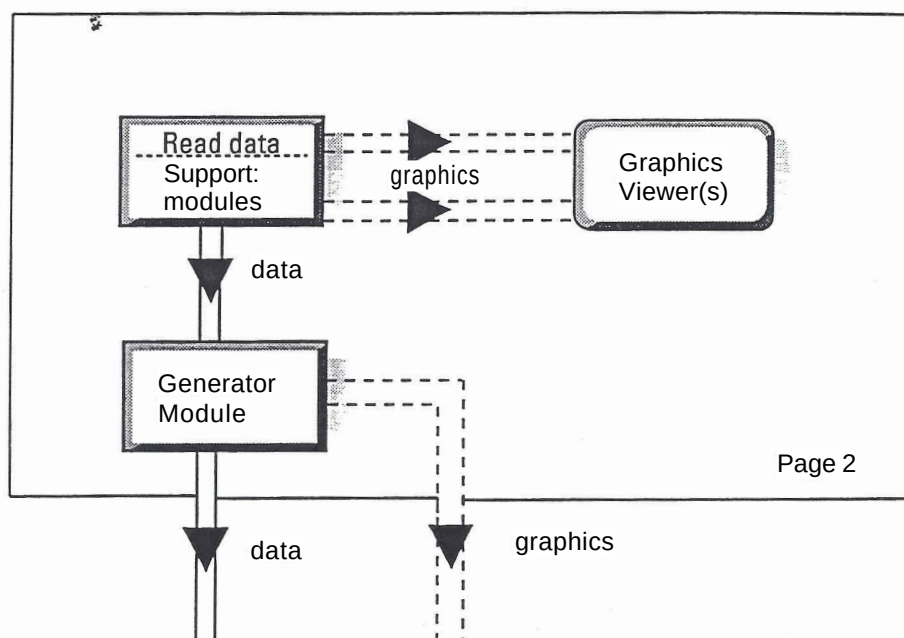


An additional feature of this design is that the generator module can write data to an output file. Thus by including support modules on Pages 2,3 and 4 that can read a file output by the generator module on the previous page

and reconstruct its graphics output, MACRO-AFFINITY can be started at any page (Figure 3.4). Thus there is a multi-level input and output functionality.

This design allows the user to apply GUI and visualization tools to specify part of the overall pollutant migration problem, with the visualization tools providing immediate information and feedback on actions taken. The following subsections provide more detail on the implementation of this design on each of the four pages. The support-generator approach and the implementation of the design principles are described in detail for Page 1. A summary is given for the other pages.

Figure 3.4:
Schematic of the
functionality of
read data support
module on Page 2.

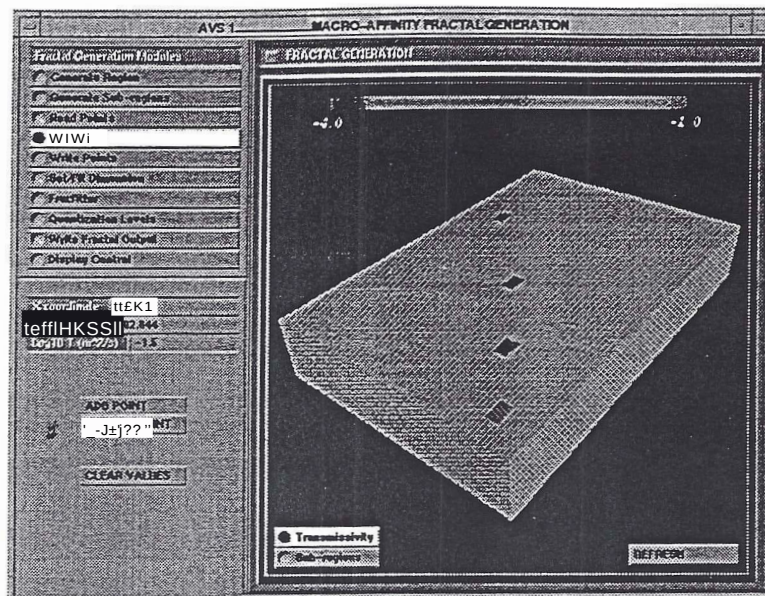


3.3 Page1: Specification of Model Region

The output from the first generator module is a permeability map that is constructed using information on the model region bounds, sub-regions, the spatial variability in each sub-region, and point-wise permeability measurements.

The first module in the support module hierarchy allows the user to specify the Cartesian (x,y) coordinate bounds for a rectangular model region, and its thickness. This generates a three-dimensional block representation of the model region on the graphics viewer on Page 1 (Figure 3.5). The reason for using a three-dimensional representation is to highlight that the thickness of the model region is important (since it affects the speed of pollutant migration) and, when the model region is a geological strata, is of physical significance. The model region is initially uniformly coloured grey.

Figure 3.5:
MACRO-AFFINITY
Page 1, showing
permeability data in
the model region.



The next support modules in the hierarchy allow the user to independently define the numerical grid on which the permeability map will be generated, define sub-regions, and import/manipulate pointwise permeability measurements.

The numerical grid generated is a simple rectilinear grid. The generator module requires only two parameters, namely the number of grid cells in the x -coordinate direction, and the number in the y -coordinate direction. The module has an option to overlay the grid mesh (formed by the grid cell boundaries) on the model representation. The sub-region module allows the user to specify sub-regions of specified shape (rectangular, circular,...) and orientation in the model region. The module generates a wire-frame of the sub-region boundaries which it places over the model region representation and the grid mesh. Since a circular sub-region cannot be mapped exactly on a rectilinear grid the user has an option to view how the sub-regions are mapped (discretized) onto the numerical grid, if it has been defined. Every grid cell is colour-coded according to the sub-region in which they are situated (determined by the position of the grid cell centre).

A group of support modules allow the user to import text information on permeability measurements and interact with the permeability data set. The permeability data set is optionally displayed on the model region representation by colour coding grid cells in which data is located (Figure 3.5). To avoid confusion the permeability data and sub-region colour-coding options are mutually exclusive. The data set can be examined by mouse point-and-clicks. Clicking on a colour-coded grid cell, the location and value of the data point within the cell are displayed in type-in widgets on the module sub-page. Existing data points can be modified or new ones added by clicking on a grid cell and entering the desired permeability value in the type-in widget. This facility is particularly important in what-if calculations, since it allows the user to construct their own

interpretation of data.

A final point of note here is that the visualization of the data is based on colour-coding numerical grid cells, since the permeability map requires cell based data. Each data point however has an exact (x, y) location, independent of the numerical grid. One of the support modules is thus responsible maintaining a list of the (a, y) locations and data values, and mapping these onto the grid. If the grid is changed the module executes again to compute the new mapping. This is one example of the responsibility of modules for local data structures and the independence between numerical and visualization structures.

The final support module in the hierarchy allows the user to set parameters that for the geostatistical method which is used to generate a permeability map, interpolates the permeability data set. The geostatistical method utilizes statistically **self-affine fractals** to generate realistic spatially variable permeability maps and requires at least three parameters: mean; variance; and fractal dimension, for each sub-region [4]. Clicking the mouse in a sub-region, the user is presented with a radio-button widget listing the options for the type of fractal used in generating the permeability map. When one option is selected, type-in widgets are displayed to indicate what parameter values need to be set for the chosen option. A diagnostics tool is also available, producing line-graphs constructed from the permeability data set to assist the user in choosing appropriate parameter values. The display of this line graph is controlled so that if the user selects a different sub-region, the diagnostic line-graph from the original sub-region is removed.

The use of the mouse to interact with the permeability data set and select sub-regions for the setting of geostatistical parameters could lead to possible conflicts. First, the user needs to be made aware of the functionality of the mouse, and secondly, different modules need to know how to process information coming from mouse clicks in a graphics viewer. The first conflict is resolved by use of the module stack. The user can only change the mouse functionality by selecting specific modules from the stack index. In addition, the functionality at a given time is clearly indicated by the one module sub-page currently visible in the module stack. The conflict over how modules react to mouse-clicks is resolved by each module keeping a note of its position in the module stack. If it is not at the top of the stack it will not respond to mouse clicks. This is an example of the design principle that modules are responsible for checking when and how they execute.

As noted above, the generator module for Page 1 uses data from each of the support modules to generate a permeability map. The generator module is listed last in the module stack index and has only one user-accessible widget, an "Execute" button, and a table (which cannot be overtyped) listing the data available. This is intended to highlight to the user that the module is the final destination for data from support modules and should be the last module to be

executed on Page 1.

On execution, the permeability map generator checks that the input data is suitable. If it is not, execution is halted and a diagnostic message is displayed. Otherwise the input data is passed to a numerical subroutine in the module. The subroutine returns a numerical data structure to the module, which is then converted into a data structure suitable for display in the graphics viewer in Page 2 and/or output to a file. The use of different data structures for the numerical permeability map generator and the visualization of results may at first sight seem inefficient. In fact, the increase in efficiency gained from using the most suitable data structure within the numerical subroutine, both in terms of GPU and development time more than offsets the conversion time. Additionally, if the visualization requirements are changed, only the conversion part of the module needs modification. This is the independence between numerical and visualization data structures referred to in the implementation-level design principles listed in Section 3.1.

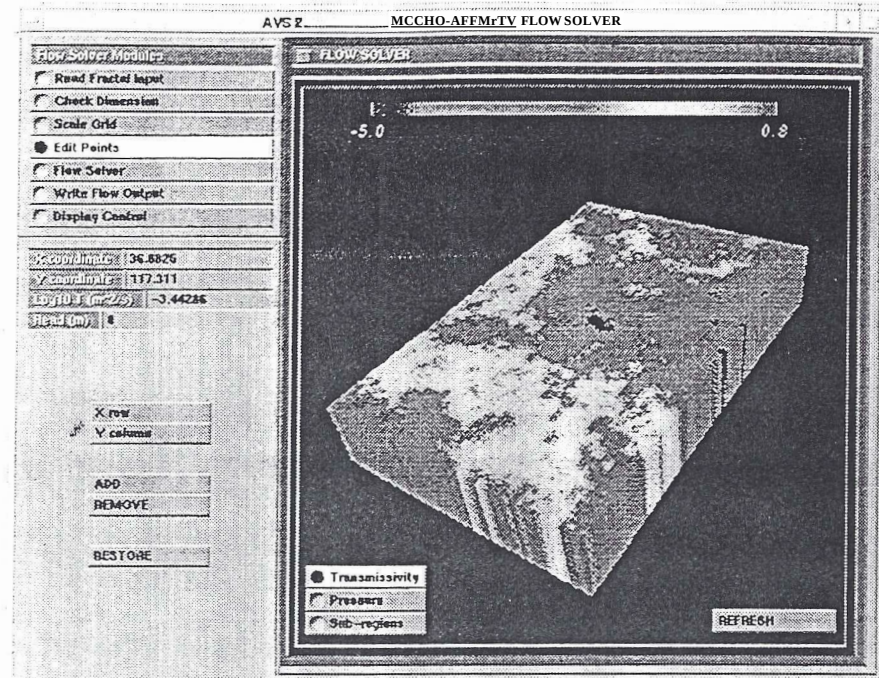
3.4 Page 2: Specification and Solution of the Groundwater Flow Equations

The generator module on Page 2 is a module containing the numerical subroutine solving the groundwater flow equations. Support modules enable the user to:

- A subdivide the numerical grid used by the permeability map generator in order to provide a new numerical grid for the flow solver;
- A specify pressure conditions at points in the model region or along the edges of the model region;
- A examine and modify the permeability map;
- ▲ set the convergence criteria parameter.

The graphics viewer on Page 2 initially contains a three-dimensional block representation of the model region with numerical grid cells colour-coded by permeability (Figure 3.6). A widget in the graphics viewer allows the user to change the colour coding to show the set pressure conditions or the sub-regions. Again this visualization is cell-based, rather than node-based. A legend for the colour coding is automatically updated depending on the data being displayed. Changes to the permeability map and set pressure conditions can be made in the same manner as changes to the permeability data set used by the permeability map generator on Page 1.

Figure 3.6:
MACRO-AFFINITY
Page 2, showing the
permeability map.



On execution of the generator module, a check is run on the data provided by the support modules. Provided these checks are satisfied, the generator module executes the numerical solver subroutine (which again uses data structures independent from the visualization data structures) and then converts data on the permeability map, the computed pressure, and the computed velocity magnitude in each numerical grid cell, to a data structure suitable for cell-based visualization. The use of three scalar quantities (f_c , p , $|u|$) rather than two scalars (h , p) and a vector (u) simplifies visualization since each of the scalars can be treated in exactly the same way, but currently some thought is being given to a vector representation of the velocity vector field.

3.5 Page 3: Specification and solution of the pollutant migration equations

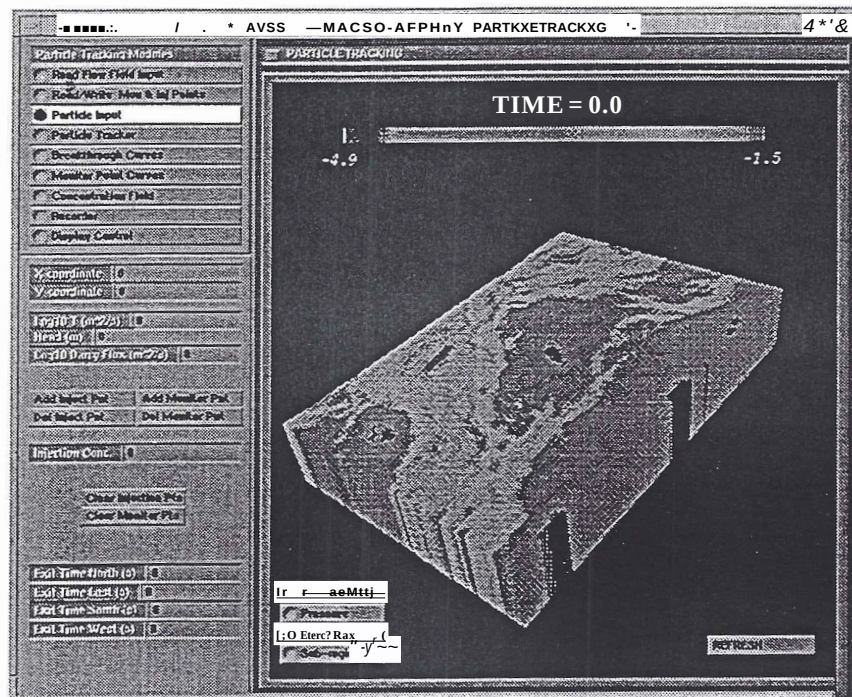
The generator module on Page 3 is a module recording an animated series of results from the numerical solver of the pollutant migration equations. This generator module is different from the previous two generator modules in that it does not directly provide visualization data structures to the graphics viewer on the following page. Instead it passes animation images to a file which can be read and visualized in the graphics viewer using a module on the following page. This allows the user to renew and compare previously run calculations.

The support modules allow the user to:

- A define pollutant source points by mouse point-and-click operations;

- A define pollutant monitor points by mouse point-and-click operations;
- A define the parameters for a particle tracking numerical subroutine which solves the pollutant migration equations;
- ▲ execute the particle tracking algorithm to generate an animated visualization of the temporal evolution of the pollutant plume;
- A construct diagnostic line graphs.

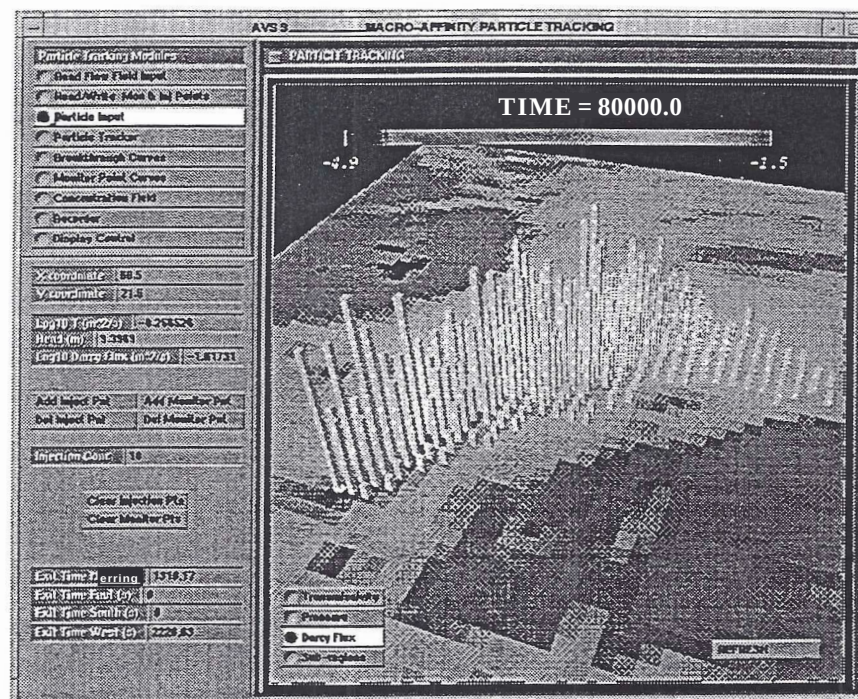
Figure 3.7:
MACRO-AFFINITY
Page 3, showing
the groundwater
velocity vector
magnitude.



From a visualization perspective, the particle tracking animation is of most interest. Initially the graphics viewer contains a three-dimensional block representation of the model region with the numerical grid cells colour-coded by either permeability, pressure or velocity vector magnitude (Figure 3.7). This cell-based visualization is constructed from the visualization data structures output by the generator module on Page 2. The particle tracking algorithm is based on a model for the movement of packets of pollutant through the numerical grid cells. The results are thus cell-based and discrete, and this has to be reflected in the visualization. In addition there is a strong link between the groundwater flow solution and the pollutant migration (for example, fast flow channels are often the primary route for migration) so the user should be able to directly compare the pollutant migration paths with the flow velocity solution and possibly the permeability map or pressure solution. The visualization approach adopted is to construct a three-dimensional cell-based histogram overlaid on the model region representation, with the height of the histogram column in a numerical grid cell being proportional to the number of particles in the cell (Figure 3.8). This histogram is updated at a user-specified time in-

terval as the particle tracking algorithm progresses. Features of the resulting animation are set by a number of parameters, such as the constant controlling the vertical scale of the histogram. The aim is to allow the user to quickly generate a series of animations until they find the viewing angle, time-step, vertical scale etc. which provides the clearest images of the migration of the pollutant through the model region. These animation images can then be recorded to file by executing the generator module.

Figure 3.8:
MACRO-AFFINITY
Page 3, showing a
detailed view of the
pollutant plume
overlaid on the
groundwater
velocity magnitude.

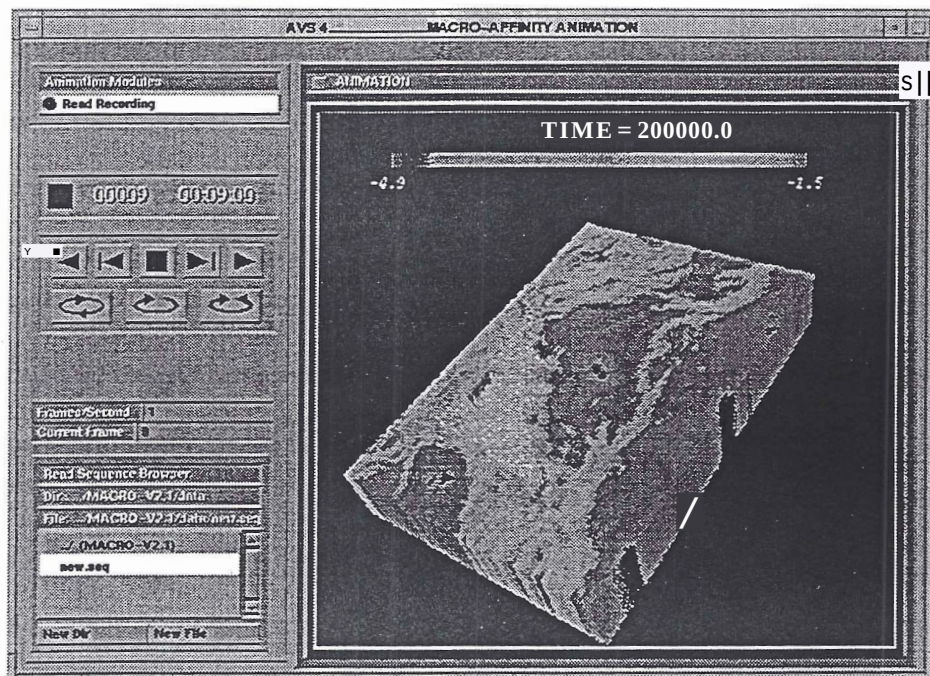


The three-dimensional histogram provides a very clear indication of the temporal evolution of the pollutant plume, but is somewhat qualitative. The use of monitor points allows the user to gather quantitative information in a manner which is directly analogous to taking field or experimental samples at selected points. An arbitrary number of monitor points can be set by clicking the mouse on the model region representation. Their set positions are denoted by coloured markers. When the particle tracker is executed the temporal variation of the concentration at each point is recorded. The result can either be written to a text file for future analysis or be displayed dynamically in a multiple line graph in an additional graphics viewer as the particle tracking algorithm executes. Additional useful quantitative information, is obtained from breakthrough curves, which are constructed by computing the amount of pollutant that crosses given model region boundaries as a function of time. The breakthrough curve data is handled and can be viewed in a similar manner to the monitor point data.

3.6 Page 4: Review of pollutant migration results

The final page of MACRO-AFFINITY is the simplest. It consists of two modules. The support module allows the user to select a file containing animation images from a pollutant migration calculation. The generator module displays the images in a graphics viewer, with "tape-recorder" style widgets allowing the user to play the animation sequence forwards, backwards or step through the sequence one frame at a time. This detailed, controllable view of the pollutant migration results together with the quantitative results provided by monitor point and breakthrough curve data is precisely the information that is required in assessing the impact of the pollutant on the environment.

Figure 3.9:
MACRO-AFFINITY
Page 4, showing
one frame in a
recorded animation
sequence.



4 Implementation of Design

4.1 Introduction

The requirements posed by a system such as MACRO-AFFINITY indicate the use of high level re-usable components, rather than a traditional, direct programming approach. The Application Visualization System (AVS) system provides a set of components and a development environment suited to the implementation and integration of analysis programs with visualization and interaction.

AVS is a 2D and 3D multi-platform, data visualization environment, based on data flow and control interfaces between synchronous or asynchronous pro-

cesses. It is used for data visualization and as a development system for visual applications. As such, it is a tool, where the appropriate parts may be used by an application, while irrelevant parts are left untouched in the base library. The AVS system serves as a foundation layer for interprocess communication, 3D graphics, user interface, and module development, while libraries of functional modules provide high level application components for data manipulation. Visual application development attempts to solve two issues, first to shorten the time from prototype to completion, and second, to provide a rich suite of tools or application components to be used during application construction.

4.2 AVS Development Framework

The primary interface to AVS is through the network editor. The network editor is a visual programming interface, allowing graphical construction of applications by making mouse directed connections between iconic program components. This visual representation is called a visualization network, or simulation network, and controls the data and execution flow in the application, through control of the flow executive.

User developed or system supplied modules are shown as small boxes in a scrollable library area, with coloured ports on the top and bottom, representing input, output, and control parameter data structures. Modules are chosen for use by dragging the boxes from the library area to the network editor construction area. Modules are then connected together into functional networks by clicking and dragging on the coloured ports. The network editor type-checks the allowable connections between modules based on data type.

The AVS system includes over 250 modules for input/output, data processing and display. The module libraries for scientific visualization are used for filtering and creating visual representations of data. Viewers, responsible for creating visible windows on data, receive representations from other modules in the network, and are responsible for handling mouse events, window system refreshes, or other display related events. In addition to the standard modules, a public domain library, maintained by the International AVS Center (<http://iac.ncsc.org>) has over 700 additional user contributed modules.

The integration of numerical solver routines, multi-level input and output, and data structure conversion, is accomplished within AVS by creating new modules. The AVS system includes program callable interfaces which enable externally supplied subroutines to be "wrapped" as AVS modules. The principle task involved in such wrapping design is the use of standard AVS prescribed data structures for input and output. Also, control parameters and a default set of user interface widgets may be specified for user supplied modules.

To assist with new module creation, AVS includes an interactive module generator, which given choice selections on its user interface panels, will generate

C source code and makefiles for new user supplied modules. The automatically generated code is a template interface, where additional source lines are added by the programmer, such as subroutine calls to numerical processing libraries.

4.3 Data Flow: Modules and Network

4.3.1 Flow Executive

The purpose of constructing a network of modules is to provide a data processing pipeline in which, at each step, the output of one module becomes the input of another. In this way, data can enter AVS, flow through the modules of a network, and finally be rendered on a display or stored outside of AVS. This process requires that each module in the network be invoked at the appropriate time.

Overall network control in AVS is the responsibility of the AVS flow executive. As such, the flow executive manages the execution order of modules. The flow executive also supervises data flow between modules, keeping track of where data is to be sent, using a variety of transport methods. The flow executive is an implementation of a data flow architecture. This differs from standard monolithic applications in that data flow does not have a global memory space or program counter. An operation in data flow (i.e. a module) is enabled if and only if all the required input values have been computed. Enabled modules consume input values and produce output values. An application based on the data flow architecture does not introduce sequencing constraints other than those imposed by data dependencies.

AVS has extended the basic data flow architecture in a form called "Adaptive Flow Architecture". Adaptive flow is optimized for a multi-process computing environment, and includes direct module data communication pipes, upstream or feedback paths, shared memory configurations, inplace memory operators, and reentrant module execution.

4.3.2 Data Flow Control Methods

Networks of modules are generally constructed interactively using the network editor. Fragments of networks can be saved as macros, or separate files, for merging into a larger application. Once a network is established, any change of data or parameter in that network is evaluated by the flow executive, which will direct the appropriate modules to respond to the change.

However, in managing complex applications, parts of the network can be created and deleted dynamically to optimize memory, or other resources. In this case

the network is only partially constructed at the initial state of the application, and the remainder is constructed by the application itself.

Three independent methods are available to control the state of an AVS network:

- ▲ mouse interaction with the network editor work area;
- jk* typing commands at the interactive command prompt (command language interpreter, or CLI);
- A seizing messages to the flow executive from within the program code of any running module.

The MACRO-AFFINITY system has used this dynamic control of this AVS mechanism to efficiently manage the flow of control and data to achieve:

- A dynamic menu user interface presentation to guide users through analysis;
- A modification of the application network to accommodate user requests;
- A signal control paths directly between processing functions as a global parameter space.

The creation of feedback loops in the network topology is also possible, and extremely useful for direct data interaction. The output from downstream modules is fed back into optional secondary inputs of upstream modules, such that a loop is formed. By data conditional behavior and parameter notifications, a series of processing steps may iterate over one data set, multiple data sets processed through a pipeline of operators, or data editing operations performed.

To create a data editing system, 3D mouse interaction with a graphics object is sent from the output of the "Viewer" module to the input of a module responsible for the supply of the data. Under appropriate conditions, data may be selected, edited, added, deleted, or placed in the view under indirect control of an edit controller module, outside of the graphics display system. The MACRO-AFFINITY system uses this facility extensively to provide computational steering, model specification, and interrogation.

4.4 Panels and Widgets

Every computational module in AVS allows for its parameters to be associated with interactive user interface elements. Floating point values may be controlled with dial, slider, or numeric text interfaces. Choice lists, radio-boxes, one-shot or toggle buttons are used for mode or choice selection parameters.

User interface elements may be grouped and organized visually onto panels, scrolled areas, or stacks. These may be placed, framed, and titled using the AVS layout editor, or by procedural control through the CLI. Layouts may be constructed by hand using the interactive placement tools, then controlled dynamically through the application program. An example of this would be to control a special options panel to appear at a certain point during data processing. Message or other dialog boxes are generally used only through program interfaces, due to their transient nature.

Interfaces can be moved, re-sized, or re-parented during program execution to create specialized user interface arrangements. User interfaces in common use withifi.an application may be stored as macros and reused in a number of distinct contexts to maintain a common look to the interface.

Specialized interfaces may also be created using external user interface design tools, or direct X/Motif programming. This allows for a number of strategies, including:

- A hiding the native AVS interface, replacing it with Motif;
- A reparenting 2D and 3D display windows into Motif interface;
- A direct X-window drawing or event handling on AVS interface;
- ▲ extending AVS interface with special Motif interaction widgets.

5 Conclusions

We have discussed the growing demand for quantitative assessment of the impact of pollutants in the environment, and identified the communication of modelling results to non-specialists and decision-makers as a key problem in meeting this demand. We have argued that integrated/interactive modelling systems provide a solution to this problem, particularly if visualization tools are used to provide users with direct access to input and output data. The commitment to such a modelling system has major implications for design, data management and performance. In this paper we have focused on design issues, and outlined a number of design principles which have been adopted in the development of the application MACRO-AFFINITY.

The top-level design principle is to exploit GUI and visualization tools to allow a user to undertake the steps of problem specification, solution and diagnostics by direct graphical interaction with results from previous steps. The lower-level design principles we have adopted are:

- ▲ divide problem into identifiable and manageable sections;

- A** adopt a molecular approach;
- A** utilize third-party GUI software;
- z!** embed numerical solvers in modules, maintaining independence between numerical and visualization data structures;
- A** allow multi-level input and output;
- A** modules are responsible for checking if they have sufficient and/or correct input data to run;
- A** modules are responsible for local data structures.

The use of third-party GUI and visualization software (such as AVS) by application developers (such as Intera) is a key point, and gives rise to many issues. Developers are seeking:

- A** visualization tools that are or can be incorporated in a development framework;
- A** quantitative, scientific visualization tools with control over details such as axes, labels, contouring and colour scales;
- A** access to visualization data structures to enable direct graphical interaction by the user and feedback from the modelling system.

In turn, application developers have a responsibility to visualization specialists to ensure that:

- A** visualization tools are used in a reasonable and appropriate manner;
- A** detailed consideration is given to the design of modelling systems incorporating visualization visualization tools;
- A** reasonable provision is made for further development of visualization tools, for example by maintaining independence between numerical sub-routines and visualization tools.

The success of MACRO-AFFINITY demonstrates that the design principles and requirements listed above provide a sound basis for cooperation between application developers and visualization specialists. Future developments will build on this to tackle the data management and performance issues that still remain if integrated modelling/visualization systems are to become more widely available.

References

- [1] M.D. Impey, M.J. Williams, J.P. Humm, S. Shah and K.J. Clark. The MACRO-AFFINITY code (MACRO-AFFINITY Version 2.0): user guide and test case. Intera Report ID3618-2 Version 4, March 1995.
- [2] S.R. Crompton, P. Grinrod, H. Takase and J. Woods. Multibarrier geochemical problems in two spatial dimensions. Submitted to *Geochem. et Cosmichem. Acta*, 1995.
- [3] P.C. Robinson and C.P. Jackson. A numerical study of various algorithms related to the preconditioned conjugate gradient method. *Int. J.P. Humm. Mech. Eng.*, 21, 1315, 1985.
- [4] P. Grinrod and M.D. Impey. Channelling and Fickian dispersion in fractal simulated porous media. *Wat. Res. Research*, 29, 4077, 1993.