A DISCUSSION OF THE VISUALIZATION NEEDS OF THE COMMUNITY - WHERE DO WE GO NEXT?

Alison M. Eyth *
UNC-Chapel Hill, Carolina Environmental Program, Chapel Hill, NC
e-mail: **eyth@unc.edu**
Web address: **http://www.cep.unc.edu/**
Voice (919) 966-2134 Fax (919) 843-3113

1. INTRODUCTION

A variety of visualization and analysis tools are used in the community today to view and analyze data related to air quality modeling. However, it seems that there is no one tool that meets all of the needs of air quality modelers. In an ideal world, air quality model inputs, emission inventories, meteorological and air quality observations, and other related data sets such as population density, water bodies, and roads could all be displayed using the same visualization tool. This would support the creation of meaningful visualizations that would assist users with drawing meaningful conclusions regarding the relationships between these distinct but related types of data.

In this paper, the limitations of today's widely used tools are discussed and some alternatives are presented. A set of requirements for a new and improved visualization tool are proposed. Input from the community on these requirements is encouraged. After the requirements are collected, the community will need to determine the most efficient method to develop the next generation visualization tool. Possible strategies include modifying an existing tool, creating a tool from some existing and some new components, or developing a new tool "from scratch".

2.0 CURRENTLY USED TOOLS

There are a number of tools that are currently widely used in the community. These include publicly available visualization tools, commercial data analysis packages, and geographic information systems (GIS). Publicly available tools include the Package for Analysis and Visualization of Environmental data (PAVE), Vis5D, NCAR Graphics, OpenDX, GrADS, and FAST. Commercial data analysis packages include SAS, Matlab, and S-PLUS. GISs include both freely available packages such as GRASS, ArcExplorer, and OpenMap, and commercial systems such as MapInfo, ArcView, ArcGIS, and ArcIMS.

PAVE gained popularity in the 1990s because it provided a quick and easy way to produce colorful graphics of air quality model related data. PAVE's strong points are the following features:

- reads many of the data formats used as inputs and outputs to air quality models directly,
- has a nice tile plot display with data shown as colored grid cells or smoothly interpolated colors,
- tile plots can concurrently animate over time,
- integrated access to configurable time series, x-z, and y-z plots,
- subsets of large data sets can be selected and plotted,
- plots are very configurable (e.g. colors, titles, subtitles),
- formulas using standard math operations can be computed and their results plotted,
- batch and interactive modes, and
- plots can be saved as image files and animations.

Although PAVE still has all of these useful features, it has some limitations. For example, its graphics are written in X-windows and Motif, which work well on UNIX workstations, but are not very well suited to PCs. One of its biggest limitations is it was built specifically to analyze gridded data, although some limited support for point data was added in Version 2. This prevents PAVE from being used for other data types such as polygons and lines, as would be required to support analysis of emission inventories, data in Shapefiles, or data associated with multimedia models. Other limitations of PAVE include a limited set of background maps, lack of animation through

* *Corresponding author address:* Alison M. Eyth, Carolina Environmental Program, UNC-Chapel Hill, 600 Bank of America CB#6116, Chapel Hill, NC 27599

space (e.g., x, y, z), it cannot be executed over the Web, and it is difficult to compile and maintain.

Another class of tools that are used by the community to analyze data other than direct model outputs is Geographic Information Systems. GISs are good tools to use for displaying data such as census data, geopolitical backgrounds, and even emission inventories because they consist of points, lines, and polygons. GISs present multiple types of spatial data in a single map-based display by treating each data set as a layer. The colors and styles used to plot the data in each layer can be configured. Typically, users can interact directly with the maps by using their mouse to get more information, zoom, and pan. GISs facilitate the analysis of the data attributes that accompany the geographic objects. "Thematic maps" can be generated that color or size the objects on a map based on the value of a data attribute. Thus, a GIS provides a powerful visualization platform.

Although GISs are powerful, they too have some limitations when it comes to examining model data. Typically, they have little support for time-varying or 3D data such as what comes out of air quality models. Although some GISs can present "2½ D" plots by showing data on top of terrain. Most GISs do not really understand 3D data, nor do they have any support for animation. GISs need to be customized to read standard air quality model input and output formats. Also, there are few cross-platform choices for GISs in that they are usually either for PCs or Unix.

Aside from PAVE or GISs, the other commonly used tools have specific strong points. For example Vis5D is a free tool than can create 3D plots. NCAR Graphics can create bar, time series, and tile plots, among others. Some of the commercial packages are very powerful and can produce very nice plots. However, many members of the community cannot afford these packages, so there are more users of the free packages. But even with all of the free and commercial packages currently available, there is still no one tool that meets all the needs of today's air quality modelers.

3.0 REQUIREMENTS FOR A NEW TOOL

3.1 Types of Requirements

If a tool is to meet the needs of today's modelers, we must first understand what those needs are. This is called the requirements analysis phase of software design, in which we define "what" the software needs to do. Another way to address software requirements is to define use cases that define "how" a user would make use of the software. Use cases can help to add additional detail on the nuances of how the software would be used for a particular task. In this text, we choose to present the requirements from a high level functional viewpoint. In the future, use cases may be used to elicit additional details.

Since there are quite a large number of requirements for a visualization tool that would meet [almost] all the needs of today's air quality modelers, it is useful to consider the requirements to be broken down into several broad categories. The categories of requirements discussed here are:

- Data Requirements to deal with the types of data that need to be handled by the tool,
- Display Requirements to address how the types of data are to be presented by the user,
- Functional Requirements to deal with computations that should be performed by the tool and other functions that should be made available to the user,
- System Requirements to address aspects of the software such has how it runs, what types of computers it runs on, and its usability and extensibility.

2.2 Data Requirements

An important high level requirement of this tool is to support the analysis of a wide variety of types of environmental data within a single software package. Air quality and meteorological modelers are a primary target audience for this tool. To meet the needs these modelers, the following types of data should be read by the tool:

- 2D and 3D temporally varying gridded data, such as that typically used and produced by air quality and meteorological models;
- temporally varying point, line, and polygon data, such as:
  - emission inventories
  - meteorological and air quality observations (both surface and upper air)
  - flight paths with observations
  - inputs to and outputs from multimedia and water quality models
  - inputs to and outputs from irregular grid air quality models

- “GIS-style data”, such as geopolitical boundaries, water bodies, road networks, and land use;
- Digital Terrain Elevation Data (DTED);
- satellite images;
- radar and lidar data; and
- output from plume models.

Note that many of the data types require that the tool have an **intrinsic understanding of 3D and temporally varying data**. In addition to dealing with the types of data described above, the tool should also be able to read both **local and remote data sets**, and have the ability to read **very large data sets** (i.e. ones that cannot be expected to fit into the computer’s random access memory).

3.3 Display Requirements

The second broad category of requirements is the display requirements, which specify how the data is presented to the user. We can get many good ideas on how users like to see their data from existing tools, although more sophisticated views may be on the horizon. In general, the tool needs to **present diverse types of data in an integarated display**. These diverse data types include 2D and 3D gridded, point, line, polygon, satellite, and radar data. Recall that all of these data types may vary as a function of time. Additional display requirements are to support:

- round earth (global) flat earth (projected) displays,
- standard and thematic maps for points, lines, and polygons,
- visualizations appropriate to radar, satellite, and other types of images,
- 3D displays such as topography and multi-colored isosurfaces
- line contours and color filled contours
- color filled grid cells and continuously shaded gridded data,
- meteorological displays such as wind vectors, wind barbs, streamlines, skew-t plots, and symbols for observations;
- transparent colors to assist with bringing together many types of data in a single display;

Another very important and useful requirement is to provide **integrated access to time series plots**, bar chars, histograms, and other useful charts from a map-based display. Finally, the tool should support **multiple related display windows** with concurrent animation.

3.4 Functional Requirements

The third broad category of requirements is the functional requirements. Again, there are quite a large number of these requirements. It is useful to categorize them as data-oriented, computation-oriented, Graphical User Interface (GUI)-oriented, and GIS-like. The following functional requirements are data-oriented:

- animate through time and space (e.g., x, y, z, arbitrary plane),
- select and plot subsets of data sets,
- probe to inspect data values / attributes (e.g. present results in tables),
- view “slices” of 4D datasets using various 1D and 2D displays (e.g. time series plots, x-t plots, bar charts, tile plots),

The following functional requirements are computation-oriented:

- compute and plot results of mathematical formulas (e.g. percent difference between two data serts),
- perform data interpolation and plot the results (e.g. interpolate point data onto a grid using Kriging),
- compute useful statistics about data (e.g. to identify exceedances or perform model evaluation).

The following functional requirements are GUI-oriented:

- provide customizable plots (e.g. legend, colors, header, footer),
- support zooming, panning and rotating,
- provide overview map for zooming and panning,
- save/reload project so you can restart your analysis where you left off.

The following functional requirements are GIS-like:

- support many map projections and earth ellipsoids,
- query to find data values that meet a criteria and analyze the attributes of or map the resulting objects (e.g. find all pont sources emitting > 100 tons NOx/yr),
- measure distances / compute areas on a map,
- show scale of the map,
- show coordinates of the mouse pointer in lat-lon and projected coordinates,
- show data sets only at scales appropriate to that data set,
- label features with attribute data,
- show pop-up tool tips based on attribute data,
- find addresses on a map.

3.5 System Requirements

The final category of requirements is system requirements. These requirements address aspects of the software such has how it runs, what types of computers it runs on, and its usability and extensibility. System requirements can further be broken down to output-oriented and operation-oriented The following are output-oriented system requirements for this tool:

- save plots and animations to image files,
- print plots,
- facilitate creation of web pages for its outputs

The following are operation-oriented requirements:

- support both batch and interactive modes,
- run over the web and on desktops,
- run on Windows and Unix systems,
- manage memory appropriately to support the analysis of very large data sets.

In addition to the above requirements the software should be **easy to extend** (i.e., to new data displays and data formats) and **intuitive to use**.

4.0 GETTING THE TOOL WE WANT

Once the requirements for the software are defined, it is a worthwhile process to perform a survey of software that is available and may support a large percentage of these requirements. New software is being produced all the time. There are literally thousands of software development projects going on around the U.S. and the world.

While at MCNC, staff members developed a prototype called the Geographic Information System Tool that, as it exists today, is similar to ESRI's ArcExplorer. However, the original design for this tool called to build in features such as PAVE-like formulas, animation, and an understanding of 3D air quality and multimedia model data that would have set it apart from many existing GIS and mapping packages. Unfortunately, the development of this package was halted when the group moved to the Carolina Environmental Program.

A tool that bears further consideration is the SpaceTimeToolkit created at the University of Alabama at Huntsville (see http://vast.nsstc.uah.edu/SpaceTimeToolkit/). This toolkit has a sophisticated display for integrated data analyses, including flight path and LIDAR data.

One tool however, seems to stand apart from the others in its scope and flexibility. This is the Integrated Data Viewer (IDV) from Unidata (see http://unidata.ucar.edu). Unidata is well known for producing software that is very useful to the atmospheric modeling community. The Integrated Data Viewer is a Java tool based on VisAD and has been many years in-the-making. It is a very sophisticated product. It has recently been released to the public as free software and comes complete with documentation and a users support group.

The IDV supports most of the data requirements outlined above. Its supported data types include 2D and 3D temporally varying gridded data, Shapefiles, radar data, satellite images, and observations. It reads remote an local files, and intrinsically understands 3D temporally varying data.

The IDV currently supports all the specified display requirements except possibly thematic mapping - although the time series plots could be improved. The display capabilities include global and flat 2D and 3D displays, data probes, vertical profiles, time-height plots, topography, isosurfaces, skew-t plots, contours, color filled contours, color filled grid cells, continuously shaded gridded data, wind vectors, wind barbs, streamlines, and meteorological observation symbols.

The IDV also supports most functional requirements, included animation through space and time, subsetting, derived parameters, probing, zooming, panning, rotating, and integrated time series and other 1D and 2D plots. However, the IDV *may not* support many of the GIS-like functional requirements.

Finally, the IDV supports most of the system requirements. It runs on Windows and Unix computers and in a web browser. It can output visualizations as JPEG and quicktime movies, and it facilitates creation of web pages. A major drawback is that it only currently supports interactive analyses, but methods for supporting batch processing are being considered. It may be possible for this community to pitch in on the development effort to have the features most important to us added into this tool

In summary, currently used visualization and analysis tools do not meet all the needs of the community. A set of requirements for a Next Generation Visualization Tool have been outlines. The Unidata Integrated Data Viewer appears to come close to meeting these requirements. The next step would be for members of the community to try this tool to see if it is something we could build on to get the visualization tool that we have been visualizing in our heads for so long.