

A JAVA METEOROLOGICAL DATA AND TRANSPORT MODEL VISUALIZATION TOOL

Stephen Masters and Mike Moore
ENSCO, Inc., Melbourne, FL

1. INTRODUCTION

Visualization of the ever-changing state of the atmosphere has been a continuing challenge for the meteorological community. Whether from real observations taken from the thousands of reporting sites around the world, or from an analysis derived from a numerical weather prediction model, the volume and complexity of the meteorological data presents special challenges to the meteorologist and software engineer.

In addition to visualizing the analyses and forecasts, users of transport-dispersion models also wish to depict the results of their models' calculations overlaid on a high-detail map. The analyst may wish to overlay concentration or dosage isopleths from several model configurations or from completely different models. Overlaying the observations or forecast model data that drove the calculations can provide useful insight into the transport models' performance.

The increasing speed and storage capabilities of modern computers allows these high volumes of data to be displayed on many scales interactively. The many platforms and operating systems in use (often within the same organization or even the same office) made it hard to provide the visualization tools wherever they were needed.

With the introduction and growth of the Java programming language and the advances in computer networking, it has become possible to write a single application that can be utilized with little or no changes across many platforms and operating systems. The Model Visualization Module (MVM) is a powerful tool for displaying distributed weather data from many sources, as well as output from several transport-dispersion models. It is a Java application that runs successfully on multiple computer platforms.

2. MVM Capabilities

The MVM displays winds, temperatures, moisture, and other fields from many sources. These sources include:

- Global 2.5-degree NCAR/NCEP Reanalysis data (Kalnay *et. al.*, 1996) stored in an Oracle® relational data base and stored in external files
- Global NOGAPS (Hogan and Rosmond, 1991) analyses and forecasts stored in an Oracle® relational data base and stored in external files.
- Surface and upper air observations from a data base.
- Analyses from the Global Atmospheric Multi-layer Trajectory Model (GAMUT, Masters and Atchison, 2000)
- Winds from the Regional Atmospheric Mesoscale System (RAMS, Pielke, 1992)
- Transport-dispersion model trajectories
 - GAMUT
 - Short-Range Layered Atmospheric Model (SLAM)
 - HYbrid Single-Particle Lagrangian Integrated Trajectory Model (HYSPLIT, Draxler, 1991)
- Concentration isopleths from SLAM and GAMUT
- Isopleths of arbitrary data from a user-provided file.

Each of these data types may be overlaid individually or any combination on a high-resolution base map. Terrain shading based on several terrain data bases may be added. Figure 1 at the end of this report shows the MVM window with navigation and data display controls. Concentration isopleths from the GAMUT transport-dispersion model are shown. Figure 2 at the end of the report show concentration isopleths from GAMUT with 500-1250 m AGL wind barbs overlaid.

Once a set of observations has been placed on the map, it may be easily "scrolled" in time. If several layers of data are shown, they may be optionally linked such that all data scrolls together in

time. If the data layers have differing native time intervals (e.g., 6-hour global analyses, 1-hour surface observations, 24-hour transport-dispersion model dosage estimates), the layers are kept in synchronization.

Data that is provided with multiple pressure-level observations may be plotted on a Skew-T/Log-P plot. As with the map plots, the Skew-T/Log-P window may show plots from several sources. The user may compare the vertical profile from several different locations within the same gridded analysis field, or may compare a profile from NOGAPS with a nearby rawinsonde observation (see Figure 3). More than six profiles may be overlaid on a single Skew-T/Log-P plot.

Currently under development is a similar capability to plot and overlay time series of any weather data element, including surface wind and cloudiness information.

3. Java Software Development

The MVM is being developed in 100% pure Java™. The main reasons for choosing Java as the programming language for the MVM is that our customers run on a variety of computers and operating systems, and show no signs of converging towards a single platform. The use of the Java programming language allows us to write software on one platform and run it on any platform that implement a Java Virtual Machine (e.g., Windows 2000/9x/NT, Linux, Sun Solaris™, MacOS, SGI Irix). The Java programming language is also a powerful, high-level language that is simple, object oriented, robust, secure, and industry supported.

The MVM is being developed using Object Oriented Design (OOD). The chief benefit that OOD provides in developing the MVM is the ability to rapidly ingest new types of data and add new functionality. An example of this is our model interface: We were able to integrate a new model (Hysplit) into the MVM in less than a day. We were also able to allow the user to read weather data from flat file/CD-ROMs instead of a database in less than a week.

As well as Object Oriented, the MVM team practices a lightweight design methodology (eXtreme Programming). The main features of lightweight design methodologies are customer focus, adaptability, collaboration (knowledge sharing), rapid feedback and minimalism (simplicity). The features that have been the most useful for our development team are:

- *Customer focus:* Adding functionality with the most business value to the customer first.
- *Rapid feedback:* Planning and development are done in two-week cycles allowing the development team to verify they are working toward the customer's desired functionality.
- *Collaboration:* Programmers working in pairs have spread system knowledge and allowed for more innovative solutions to design problems.

4. Notes

Oracle is a registered trademark of the Oracle Corporation.

Java and Solaris are trademarks of Sun Microsystems, Inc.

5. References

- Draxler, R.R. and G.D. Hess, 1997, Description of the Hysplit_4 modeling system, *NOAA Tech Memo ERL-ARL-224*, Dec, 24p.
- Hogan, T., and T. Rosmond. 1991: The description of the Navy Operational Global Atmospheric Prediction System's spectral forecast model. *Mon. Wea. Rev.*, 119, 1786-1815.
- Kalnay, E., and coauthors, 1996: The NCEP/NCAR 40-Year Reanalysis Project. *Bull. Amer. Meteor. Soc.*, 77, 437-471.
- Masters, S.E. and M.K. Atchison, 2000. Synoptic scale vertical effluent redistribution: A comparison of two modeling approaches. *Preprints 11th Joint Conf. of the App. of Air Pol. Met. with A&WMA. 9-14 Jan 2000.* 236-240.
- Pielke, R.A., 1992. A comprehensive meteorological modeling system RAMS. *Meteor. Atmos. Phys.* 49:6991.

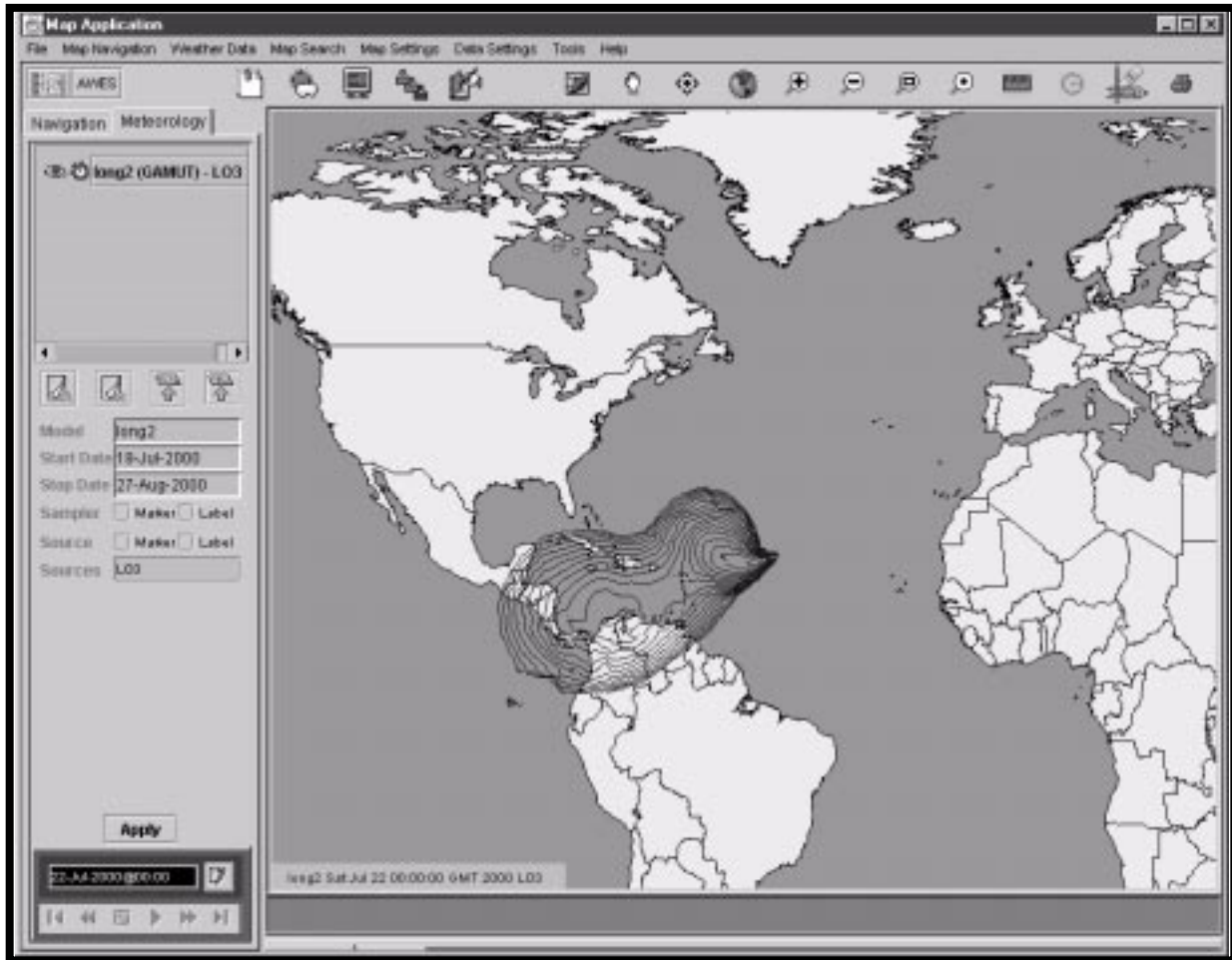


Figure 1. Full Model Visualization Module (MVM) window. Concentrations calculated using the GAMUT transport-dispersion model are shown. Along the top are map navigation controls and data selection menus. At bottom left are the time scrolling and syncing controls.

Figure 2 (below). GAMUT concentration isopleths with 500-1250 m AGL wind vectors overlaid.

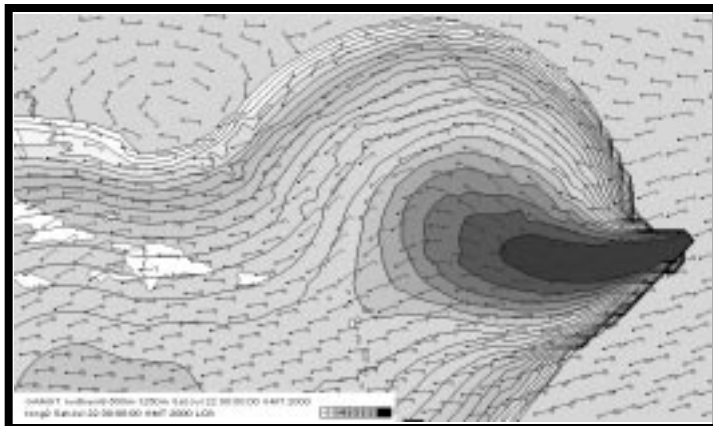


Figure 3 (right). Skew-T/Log-P plot with vertical profile from a rawinsonde sounding from Topeka, Kansas and the nearest NOGAPS grid point. Solid lines denote temperature; dashed lines dew point

