

INTERACTIVE FEDERAL STATISTICAL DATA ON THE WEB USING “NVIZN”

Jon Hurst

Utah State University, Department of Mathematics and Statistics
3900 Old Main Hill, Logan, UT 84322-3900

e-mail: jon@jonathan.hurst.name

Jürgen Symanzik

Utah State University, Department of Mathematics and Statistics
3900 Old Main Hill, Logan, UT 84322-3900

e-mail: symanzik@math.usu.edu

Lacey Gunter

University of Michigan, Department of Statistics
439 West Hall, 550 East University, Ann Arbor, MI 48109-1092

e-mail: lgunter@umich.edu

Abstract

Online applications are an attractive solution for providing quick access to geographically referenced Federal data sets. In the past, available software was not ideally suited for interactive, statistical graphics applications on the Web. “nViZn” (read envision) is a Java-based software development kit for statistical graphics. Building on the “nViZn” libraries, we developed software for the interactive display of Federal air pollution data. This software allows users to display, sort, and compare multiple tables and micromaps. Having produced this display framework, we conclude that “nViZn” based applications are a good solution for interactive statistical graphics on the Web.

1. Introduction

The research and data-gathering of many Federal agencies results in large, geographically-referenced data sets that in the past have been difficult to distribute in a usable and meaningful form. During the last five years, Federal agencies and other institutions have devoted much effort to developing Web-based applications for the distribution of their data, resulting in several good approaches to this problem. Graphical displays and Web interfaces have been developed that allow users to dynamically sort and compare geographic data on various scales and at various resolutions, thus making these large data sets much more usable. Shneiderman (1999, originally published in 1994) writes, “The dynamic query approach lets users rapidly, safely, and even playfully explore a database.”

Unfortunately, at present, very few geographically referenced data sets have been made available on the Web in an interactive and graphical form. Shneiderman (1999) also concludes that “The dynamic query approach is poorly matched with current hardware and software systems.” Although the current lack of dynamic query systems that produce graphical output certainly supports Shneiderman’s observation, the situation has changed

considerably during nearly a decade since the publication of his original paper: current hardware and software systems are very capable of dynamic queries and graphical output.

In an attempt to address the challenges of large, geographically-referenced, Federal data sets, we have created a prototype application for viewing air pollution data. It allows users to dynamically create, modify, and compare data in graphical and tabular formats. Micromaps (small maps linked to statistical data) are employed to present the data as clearly as possible. Our application is based on the commercial “nViZn” visualization libraries. These libraries were used to decrease development time and due to their support of many statistical and graphical features, including micromaps. This application is a model for agencies and organizations wishing to distribute similar data on the Web.

Section 2 describes the development of interactive micromaps and their use on the Web. Section 3 describes our implementation of the interactive micromap concept using “nViZn” software and Federal air pollution data. Section 4 closes by discussing the shortcomings and possibilities of our implementation.

2. Background—Interactive, Online Graphics

Currently most Federal data available on the Web can only be accessed as long, static tables, which are usually not an effective means of understanding large data sets, especially geographically-referenced data sets. Realizing the need for more usable data delivery, the National Science Foundation and cooperating Federal agencies launched the Digital Government program in 1999 (<http://www.diggov.org>).

This program’s purpose is “to explore and develop new information technologies that will improve the way government serves the American people.” (<http://www.diggov.org>) One branch of this program is the Digital Government Quality Graphics initiative (<http://www.geovista.psu.edu/grants/dg-gg>). Its goal is dissemination of Federal statistical data on the Web in more usable and understandable forms. Funding from these efforts has helped to develop better methods for accessing Federal data online.

During the last several years the idea of linked micromap plots (often simply called micromaps) has been developed (Carr and Pierson, 1996; Carr et al., 1998, 2000). Micromaps are an excellent medium for the presentation of geographically-referenced data. Micromaps use multiple small maps to provide geographic reference for the accompanying statistical data. By dividing the data between several small maps and sorting the maps according to the statistical data, the viewer’s attention can easily focus on the data’s significance both geographically and between data points.

Micromap use on the Web was first considered by the Environmental Protection Agency (EPA) under their Cumulative Exposure Project (CEP). The original CEP goal was to provide easy, interactive, online access to their hazardous air pollutant (HAP) data through their Web site (<http://www.epa.gov/cumulativeexposure>). Unfortunately, no part of the interactive CEP Web site was ever published due to concerns that the 1990 data was outdated (Symanzik et al., 1999, 2000). Their Web site has not been updated for several years, and no data is accessible.

Currently, the U.S. Department of Agriculture-National Agricultural Statistics Service (USDA-NASS) research and development division uses interactive micromaps to display data from the 1997 Census of Agriculture (<http://www.nass.usda.gov/research/sumpant.htm>). Users can sort each micromap display by acreage or yield with respect to a selected crop. However, these micromaps are jpeg images with pre-calculated data rep-

resentations: it is not possible for a user to create new micromaps, and updated data will require the generation of new images.

The National Cancer Institute recently released a Web site (<http://statecancerprofiles.cancer.gov/micromaps>) that provides interactive access to its cancer data via micromaps. This Web site is Java-based and allows one to dynamically create micromaps (Wong et al., 2002).

3. “nViZn” Micromap Prototype Application

3.1 Introduction

As part of the Digital Government Quality Graphics initiative, our goal was to create proof-of-concept software for 1) easy access and 2) concise display of the CEP data in 3) a format usable and understandable to a non-statistical audience.

These goals were realized by 1) using software accessible from the Web, and 2) using “nViZn” software libraries to create 3) interactive drilldown maps, micromaps, and tables.

“nViZn” (<http://www.spss.com/nViZn/>) is a commercial, Java-based, software development kit for data visualization based on the Graphics Production Library (GPL). The ideas behind the GPL and subsequently “nViZn” are based on Leland Wilkinson’s book *The Grammar of Graphics* (1999, also see Wilkinson et al., 2000).

“nViZn” includes tools for the creation of many statistical graphics including drilldown maps and micromaps. Its distribution also includes sample code for many visualization approaches, which serve as templates for the implementation of drilldown maps, micromaps, and tables. The libraries also include capabilities for dynamic data filtering, animation, rotation, and the display of metadata, allowing easy interaction with displayed data.

We built our software around the “nViZn” libraries and, where possible, used sample code which we modified to suit our needs. Preliminary work to determine the suitability of producing micromaps, drilldown maps, and tables using the “nViZn” libraries is outlined by Jones and Symanzik (2001), Symanzik and Jones (2001), and Symanzik et al. (2002). Their code was used as a starting point to create the current prototype application.

In the remainder of this section we discuss this application first from a user’s perspective followed by an implementation and programming perspective.

3.2 User Interface & Examples

Our data is a duplicate of the geographically-linked CEP HAP data (introduced in Section 2). The EPA modeled 148 HAPs for each of the 60,803 census tracts in the continental US according to 1990 data (Rosenbaum et al., 1999). Values are given in micrograms per meter cubed ($\mu\text{g}/\text{m}^3$). This data is easily accessed through a hierarchical data view, allowing users to view and compare the data at various resolutions. The first and coarsest view is through the drilldown map shown in Figure 1. By zooming to the desired resolution, and selecting the appropriate HAP (bottom right corner), one can visually compare pollutant levels by the relative colors of the states. From Figure 1 we can see, for instance, that the Northeast has the highest lead pollution and the Mountain West the lowest. Exact pollutant values for each state can be displayed transiently by choosing the ‘tooltip’ tool and hovering over a state or in another window by choosing the ‘meta’ tool and selecting a state.

By choosing the ‘table’ or ‘micromap’ tools and then selecting a state, the next level of resolution can be viewed. Selecting a state with these tools creates a new window with more detailed data. Figures 2 and 3 show example tables of benzene pollution in New York

and Arizona. Figures 4 and 5 show example micromap displays of benzene pollution in New Mexico and lead pollution in California. Both tables and micromaps can be sorted ascending and descending according to the six statistical functions listed on the buttons in images 2 through 5: minimum, mean, maximum, first quartile, median, and third quartile. By selecting individual data points or counties on the micromaps, more details appear in small ‘meta’ windows. The micromaps also have sliders for setting the ‘octal’, or number of counties displayed per map, on the fly. The software initially chooses the octal (between four and ten) that most evenly distributes the counties between groupings: Figures 4 and 5 show that the software has chosen an octal of seven for New Mexico and ten for California.

The octal slider can be a useful tool for quickly gathering information from the micromaps. As an example, perhaps we are interested in quickly determining the most polluted counties in a state by glancing at the topmost map in a micromap display. By sliding the octal from high to low we can remove counties from the topmost map to reveal different (or the same) patterns. This is illustrated in Figures 6 and 7, which show two partial micromaps of benzene pollution in Vermont with different octal values selected. Figure 6 shows an octal of eight and at a glance indicates that the southern and western parts of the state are the most polluted. Figure 7 shows an octal value of four and indicates that the northwestern corner is the most polluted at a finer scale.

Figure 8 shows how the desktop might look after using the software for several minutes. The drilldown map has been focused on the Mid-Atlantic States. A table of benzene pollution in New Jersey has been created to view exact county values. Two micromaps of benzene pollution in New Jersey have been created to compare the symmetry of the data: the first sorted by the mean and the second by the median. The graphs indicate that the data is fairly symmetrical. The maps show that many of the counties with highest pollution are in the vicinity of Newark. The two small ‘meta’ windows show the exact mean and median values of benzene pollution in Union county.

3.3 Technical Details

3.3.1 Software Structure & Details

Figure 9 details the structure of the software, how it interacts with the data, and how it produces output (source code is available by contacting the authors):

- `Hazmap.java` includes all of the code that handles the drilldown map, including data interaction. This code is little changed from the sample drilldown map included with the “nViZn” distribution, with the exception of the additions which create `Table` and `MicroMap` objects. When `Table` and `Micromap` objects are created, they are passed the HAP and two letter abbreviation of the state in question.
- `Table.java` includes the code for creating and interacting with `Table` objects. This class uses methods from the `StateCodes.java`, `CountyCodes.java`, and `HapCodes.java` classes to interpret the two letter state abbreviation received from `Hazmap.java` and the FIPS codes used in the directory structure of the CEP data.
- `MicroMap.java`, `mmTools.java`, and `MMDData.java` are related classes which create `MicroMap` objects. Together they contain almost 1,000 lines of code, and thus were split into three objects for manageability and to increase orthogonality between interacting methods. As with `Table.java`, the `StateCodes.java`, `CountyCodes.java`, and `HapCodes.java` classes are used for code interpreta-

tion. Additionally, `GenFileShapeReader.java` is used to acquire the spatial data needed to draw the micromaps.

- `StateCodes.java`, `CountyCodes.java`, and `HapCodes.java` create `CSVParser` objects to access the comma separated value (CSV) files containing FIPS codes for states, counties, and HAPs, thus creating an independent data access layer.
- `CSVParser.java`, `CSVParse.java`, and `CSVLexer.java` are freely available classes (available at <http://ostermiller.org/Utils/CSV.html>) licensed under the GNU General Public License that provide robust handling of CSV files.
- `GenShapeFileReader.java`, as noted above, acquires spatial data from “gen” files (see Section 3.3.2). This class is a modification of “nViZn” source code.

The “nViZn” libraries completely handle the creation of maps, graphs, and tables. In theory, once the data are in memory, the programmer only needs to indicate which variables to display, how they are displayed, and how they are sorted (Section 3.3.2 details how this is different for micromaps).

Interaction with the “nViZn” graphics is straightforward because the commands for display, formatting, and sorting can be issued at any time after the creation of the graphic. In our software this is accomplished through interaction with standard java widgets: buttons, sliders, and dialog boxes.

The code for creating and configuring the “nViZn” graphics is concise and can be written quickly. However, learning how to write this code is not obvious and required considerable time. In some instances we were unable to use certain capabilities of the “nViZn” libraries because we were unable to determine how to use them, and less efficient work-arounds were used instead. This added a great deal of development time, and could have been avoided by better documentation.

Besides the code for the use of and interaction with the “nViZn” libraries, the code for data manipulation and creation of the user interface is somewhat substantial: the entire application (not including the third party CSV classes) is about 2,000 lines of code.

3.3.2 Data Format & Handling

The CEP data is stored in space delimited, ASCII, flat files in a directory structure by state, county, and HAP. The directories and files of states and counties are referenced by numeric Federal Information Processing Standards (FIPS) codes (<http://www.itl.nist.gov/fipspubs/>), and the pollutants by numeric HAP codes.

Data access requires several steps. First, for a given state and pollutant the numeric FIPS state and county codes and HAP code are retrieved from a flat file. These codes are then used to retrieve the corresponding data from the CEP data structure. The data is then written to a temporary flat file in a format that can be recognized by the “nViZn” libraries. For micromaps, additional octal and spatial data must be added to the temporary file. Finally, the “nViZn” libraries load the data from this temporary file into memory.

A seemingly needless transfer of data from file to memory to file to memory occurs in this process. Indeed passing the data in memory directly to the “nViZn” libraries instead of writing them to a temporary file would be much more efficient. However, we could not accomplish this with the given documentation, and this work-around was used instead.

This work-around is especially problematic with micromaps. Because octal values can be changed on the fly, and octal data must be added to the temporary file, the file must be rewritten and reloaded between every sort. The data in micromaps can also be sorted by the

values given by six statistical functions. Because the octal values are relative to the sorted data, routines to sort the data according to these six sets of values also had to be written so that the octal data could be properly added before the temporary file was written.

Micromaps also require data designating the shape of each state and county. We used the shape files from the original CEP project, which were generated with ArcView as “gen” files (Symanzik et al., 2000, describes how these files were generated). The “nViZn” libraries require shape files in a custom format, however the “nViZn” developers provided us with their source code which we modified to accept ArcView gen files. This entailed redefining the formats of headers and footers associated with the spatial data, and associating the headers (which include FIPS codes in “gen” files) with their corresponding counties.

3.3.3 Using Other Data

These data handling routines could be easily modified to accept other geographically-referenced data sets. Four portions of the code must be changed, three of them quite easily:

- 1 & 2. The two (different) routines that read data from its original format into memory—one for tables and one for micromaps—must be modified. The format of the developer’s data must be known, and a routine to parse it into an array must be developed. Our data contained six pre-calculated statistical functions; for raw data (e.g., data with 100 sample points per county) these statistical functions would have to be calculated and then loaded into the array. The “nViZn” libraries can perform many statistical functions, and using these library routines may be a good approach.
3. The data must be structured like the CEP data, or the data accessing routine must be rewritten. If the data is not structured in directories according to FIPS codes, the software developer must either organize it in this way or create a comma separated value file relating state and county names to their corresponding directories. The program must then be modified to read directories according to this file.
4. If the drilldown map is to be used, its data source must be modified, which is problematic. The pollution and shape data for this map are found in one very large file whose format we have not dealt with. This would require some study and change for application to other uses.

4. Conclusion

Our prototype application greatly facilitates exploration of geographically-linked, Federal data sets (using air pollution data as a test data set) by employing the “nViZn” libraries and micromaps. Although some limitations became evident, especially the lack of “nViZn” documentation, we found the “nViZn” libraries and their direct support of micromaps to be an effective means of quickly producing an application for visualizing these types of data sets. The “nViZn” libraries have recently been updated, including additional documentation and sample code, possibly alleviating the documentation issues we experienced.

At the writing of this paper, our application lacks two notable features. First, it currently runs as a stand-alone application, not as an applet in a Web browser. However, the “nViZn” libraries—and the Java language itself—are written to accommodate applet Web applications, and this shortcoming is a matter of time and not possibility. Second, the HAP data and ArcView shape files are stored in a hierarchical file structure originally developed for the CEP site. This is certainly outdated compared to current relational databases. The “nViZn” libraries support database operations, and this support could be leveraged in future development efforts.

Acknowledgements

Jürgen Symanzik's work was supported in part by the NSF "Digital Government" (NSF 99-103) grant #EIA-9983461. Jon Hurst's and Jürgen Symanzik's work was supported by a New Faculty Research Grant from the Vice President for Research Office from Utah State University.

References

- Carr, D. B., Olsen, A. R., Courbois, J. P., Pierson, S. M., Carr, D. A., 1998. "Linked Micromap Plots: Named and Described". *Statistical Computing and Statistical Graphics Newsletter* 9 (1), 24-32.
- Carr, D. B., Olsen, A. R., Pierson, S. M., Courbois, J. P., 2000. "Using Linked Micromap Plots to Characterize Omernik Ecoregions". *Data Mining and Knowledge Discovery* 4 (1), 43-67.
- Carr, D. B., Pierson, S. M., 1996. "Emphasizing Statistical Summaries and Showing Spatial Context with Micromaps". *Statistical Computing and Statistical Graphics Newsletter* 7 (3), 16-23.
- Jones, L., Symanzik, J., 2001. "Statistical Visualization of Environmental Data on the Web using nViZn". *Computing Science and Statistics* 33, Forthcoming, (CD).
- Rosenbaum, A. S., Axelrad, D. A., Woodruff, T. J., Wei, Y.H., Ligocki, M. P., Cohen, J. P., 1999. "National Estimates of Outdoor Air Toxics Concentrations". *Journal of the Air and Waste Management Association* 49, 1138-1152.
- Shneiderman, B., 1999. "Dynamic Queries for Visual Information Seeking". In: Card, S. K., Mackinlay, J. D., Shneiderman, B. (Eds.), *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, San Francisco, CA, pp. 236-243.
- Symanzik, J., Carr, D. B., Axelrad, D. A., Wang, J., Wong, D., Woodruff, T. J., 1999. "Interactive Tables and Maps -A Glance at EPA's Cumulative Exposure Project Web Page". In: *1999 Proceedings of the Section on Statistical Graphics*. American Statistical Association, Alexandria, VA, pp. 94-99.
- Symanzik, J., Hurst, J., Gunter, L., 2002. "Recent Developments for Interactive Statistical Graphics on the Web using "nViZn"". In: *2002 Proceedings*. American Statistical Association, Alexandria, Virginia, (CD), forthcoming.
- Symanzik, J., Jones, L., 2001. "'nViZn' Federal Statistical Data on the Web". In: *2001 Proceedings*. American Statistical Association, Alexandria, VA, (CD).
- Symanzik, J., Wong, D., Wang, J., Carr, D. B., Woodruff, T. J., Axelrad, D. A., 2000. "Web-based Access and Visualization of Hazardous Air Pollutants". In: *Geographic Information Systems in Public Health: Proceedings of the Third National Conference August 18-20, 1998, San Diego, California*. Agency for Toxic Substances and Disease Registry, <http://www.atsdr.cdc.gov/GIS/conference98/>.
- Wilkinson, L., 1999. *The Grammar of Graphics*. Springer, New York, NY.
- Wilkinson, L., Rope, D. J., Carr, D. B., Rubin, M. A., 2000. "The Language of Graphics". *Journal of Computational and Graphical Statistics* 9 (3), 530-543.
- Wong, X., Chen, J. X., Carr, D. B., Bell, S. B., Pickle, L. W., 2002. "Geographic Statistics Visualization: Web-Based Linked Micromap Plots". *Computing in Science and Engineering* 4 (3), 90-94.

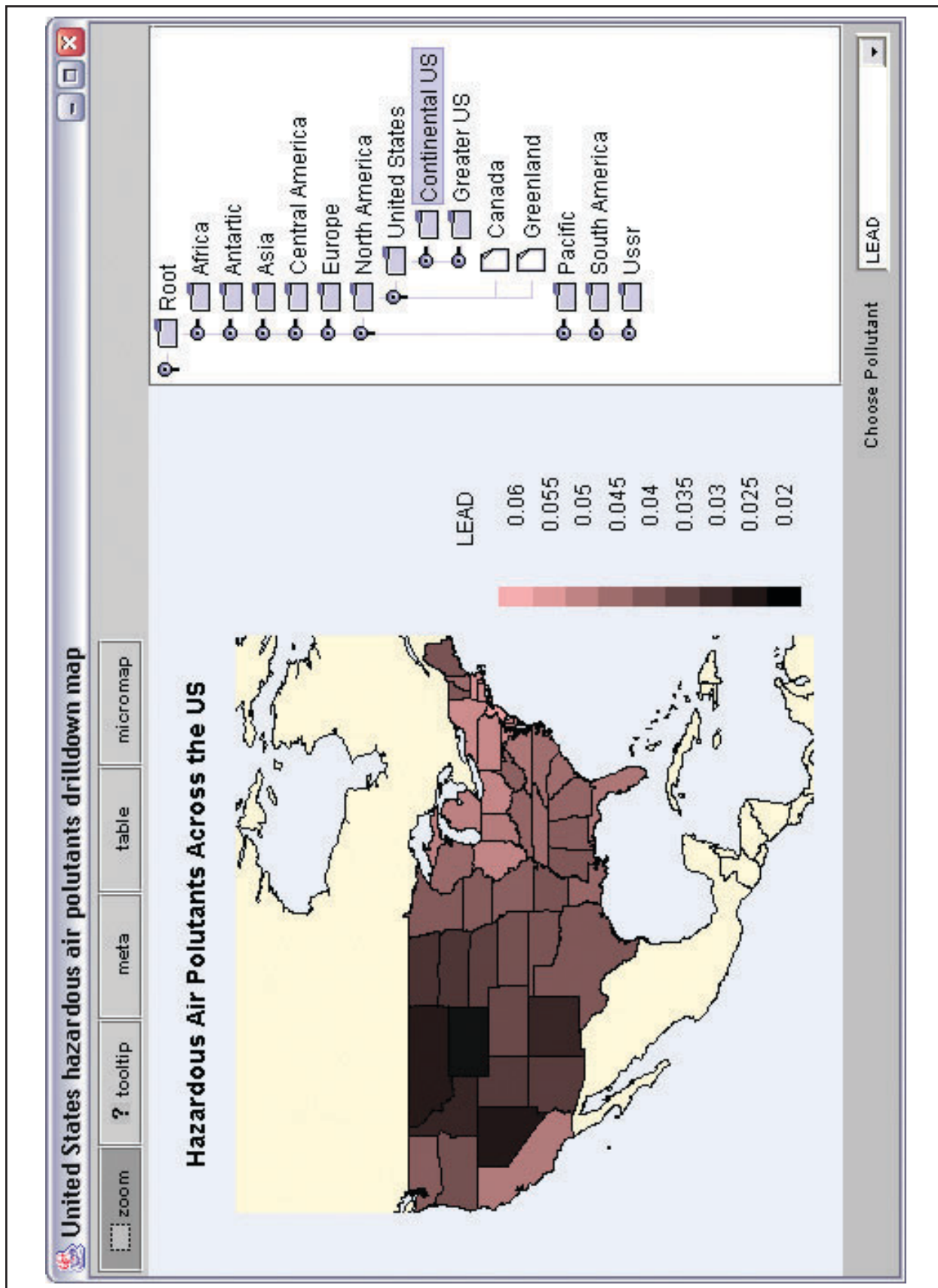


Figure 1—An example of the drilldown map showing lead pollution at the state level.

Sort table by: minimum mean maximum quartile 1 median quartile 3

Arizona 1990 Modeled benzene Concentrations

	Minimum	Mean	Maximum	Quartile1	Median	Quartile3
Maricopa	0.57	2.51	12.68	2.01	2.53	2.91
Pima	0.51	1.51	4.78	1.19	1.46	1.66
Yuma	0.49	1.33	2.91	0.66	1.24	1.9
Cocconino	0.49	1.29	3.72	0.66	1.03	1.58
Pinal	0.52	0.92	3.8	0.6	0.73	1.0
Cochise	0.51	0.79	2.28	0.52	0.66	0.92
Santa Cruz	0.5	0.72	1.19	0.51	0.65	0.87
Yavapai	0.49	0.85	3.97	0.52	0.64	0.8
Navajo	0.52	0.67	1.06	0.55	0.63	0.8
Mohave	0.48	0.8	2.11	0.51	0.63	0.83
Gila	0.51	0.7	1.1	0.52	0.59	0.73
Graham	0.49	0.78	1.84	0.53	0.56	0.7
Apache	0.52	0.57	0.72	0.53	0.54	0.6
Greenlee	0.5	0.55	0.62	0.53	0.54	0.56
La Paz	0.48	0.52	0.59	0.49	0.51	0.51

Sort table by: minimum mean maximum quartile 1 median quartile 3

New York 1990 Modeled benzene Concentrations

	Minimum	Mean	Maximum	Quartile1	Median	Quartile3
New York	3.18	6.38	13.27	5.54	5.85	6.64
Bronx	2.56	4.93	12.01	4.25	4.85	5.32
Kings	2.48	4.88	16.77	4.06	4.74	5.24
Queens	2.04	4.25	22.53	3.57	4.04	4.53
Richmond	2.03	3.6	6.68	3.18	3.58	4.05
Westchester	1.01	2.66	5.8	1.97	2.55	3.24
Nassau	1.51	2.56	6.6	2.24	2.5	2.86
Schenectady	0.78	2.45	4.33	1.62	2.48	3.21
Monroe	0.82	2.41	4.64	1.46	2.33	3.18
Albany	0.7	2.22	5.52	1.43	2.06	2.75
Onondaga	0.81	2.2	5.53	1.31	2.19	2.97
Suffolk	0.64	2.15	10.32	1.79	2.07	2.35
Rockland	1.33	2.07	3.42	1.66	1.96	2.39
Niagara	0.76	1.97	4.19	0.99	1.81	2.47
Erie	0.67	1.95	4.82	1.36	1.9	2.48
Rensselaer	0.81	1.88	5.05	1.0	1.65	2.58
Oneida	0.65	1.82	4.16	0.87	1.59	2.66
Broome	0.67	1.6	3.65	0.87	1.32	2.18
Chemung	0.66	1.58	2.6	0.92	1.74	1.97
Orange	0.75	1.49	3.84	0.89	1.23	1.89
Dutchess	0.71	1.47	4.58	0.94	1.05	1.8

Figure 2 (top) and Figure 3—Two examples of interactive tables showing benzene pollution in New York and Arizona at the county level.

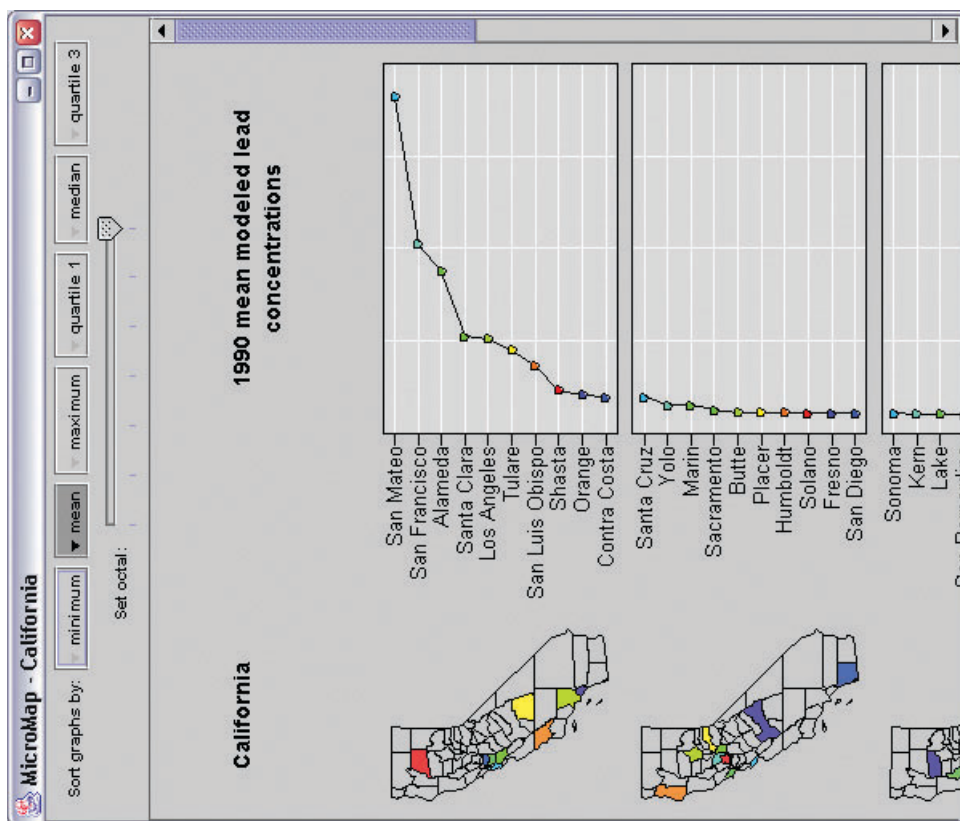
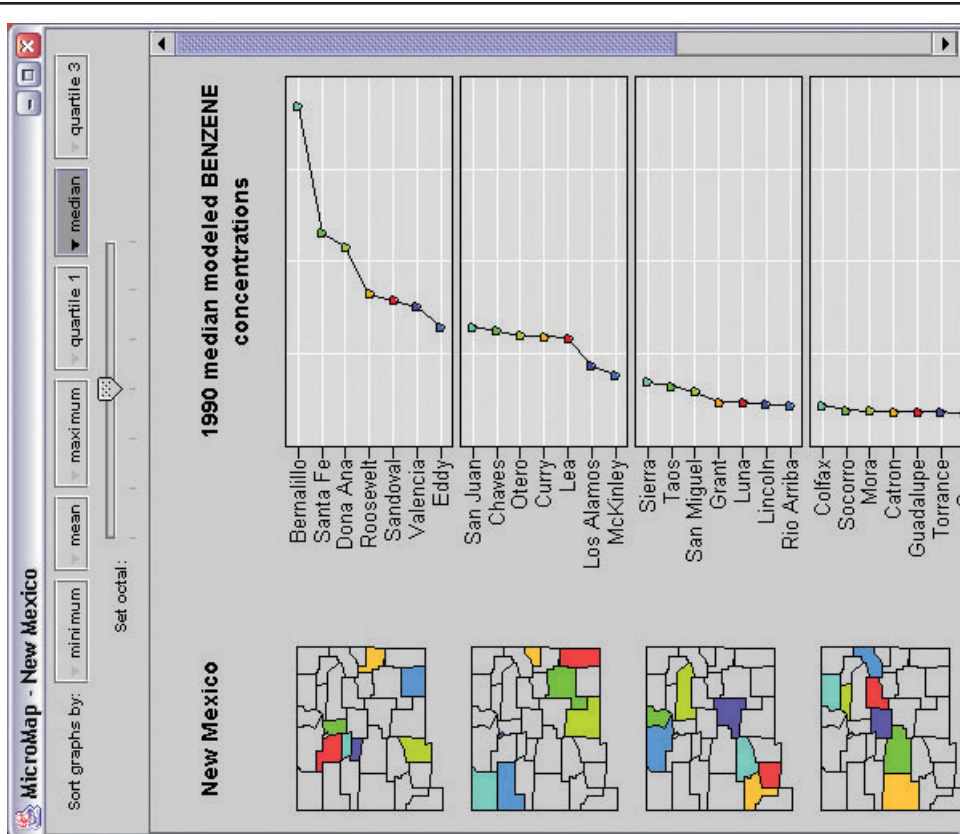


Figure 4 (top) and Figure 5—Two examples of interactive micromaps showing benzene pollution in New Mexico and lead pollution in California at the county level.

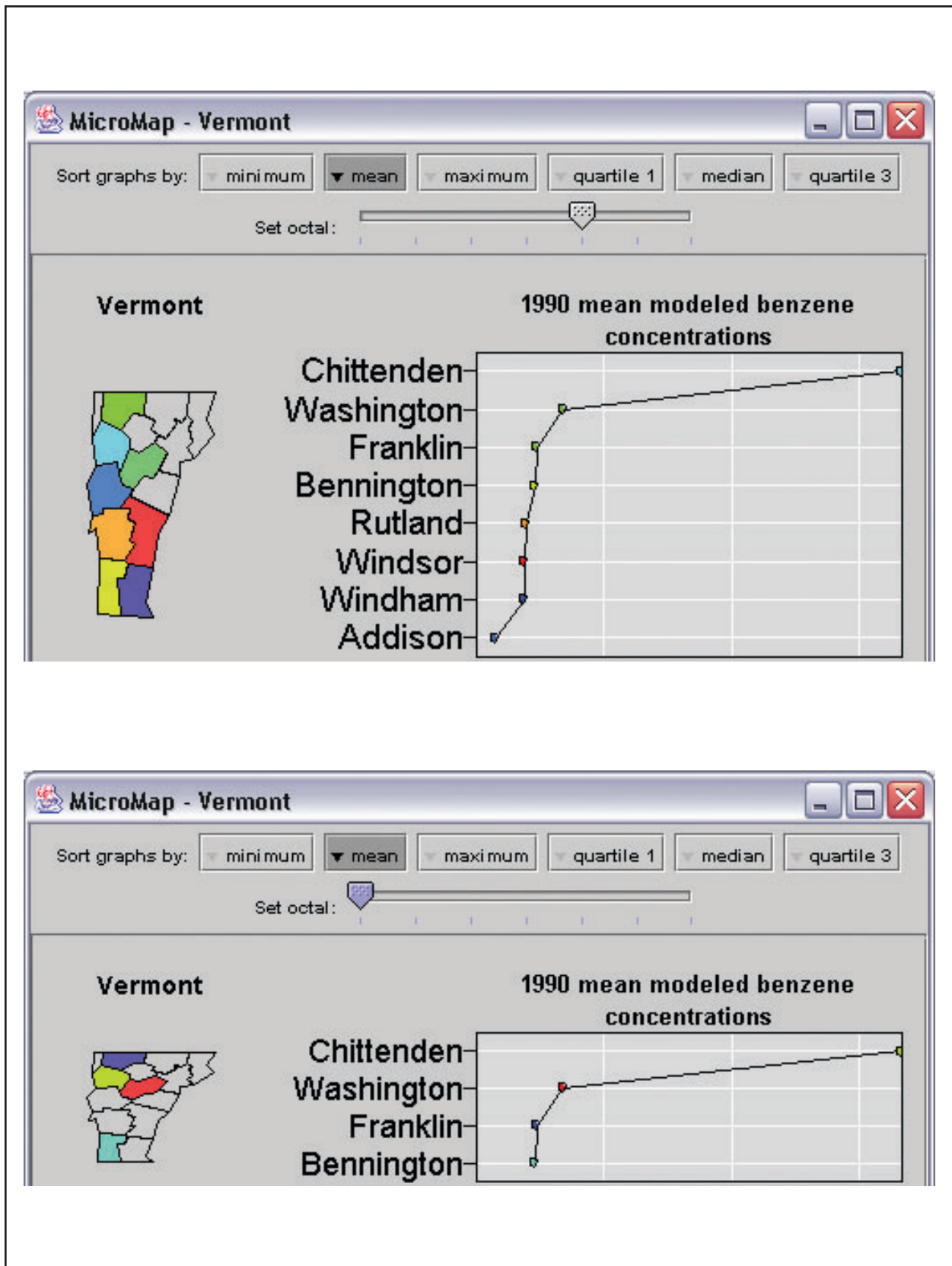


Figure 6 (top) and Figure 7—Use of the octal slider to quickly determine the most polluted counties, demonstrated with benzene pollution in Vermont.

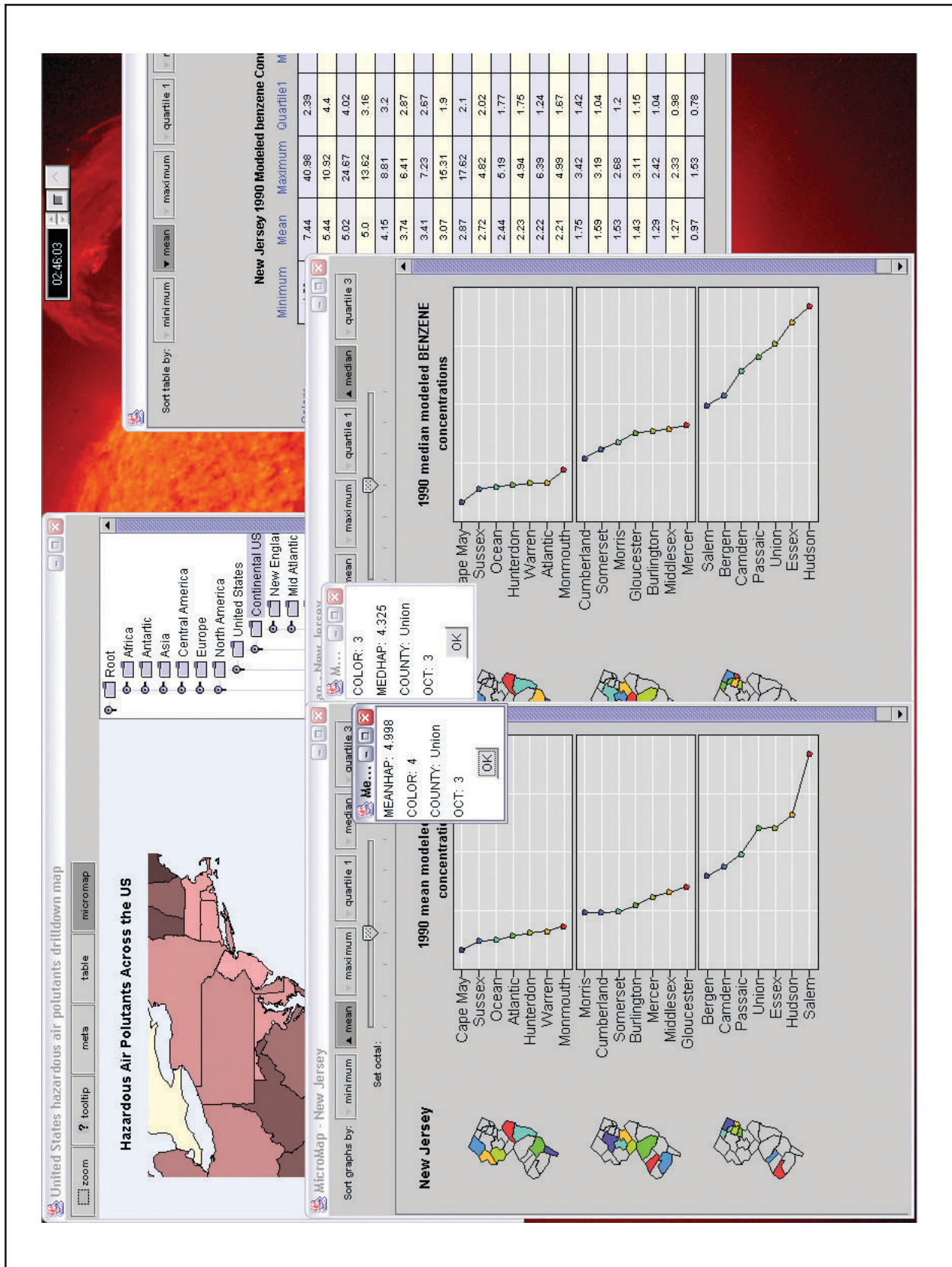


Figure 8—Desktop appearance after using the software for several minutes, showing a drilldown map, two micromaps, one table, and two meta windows.

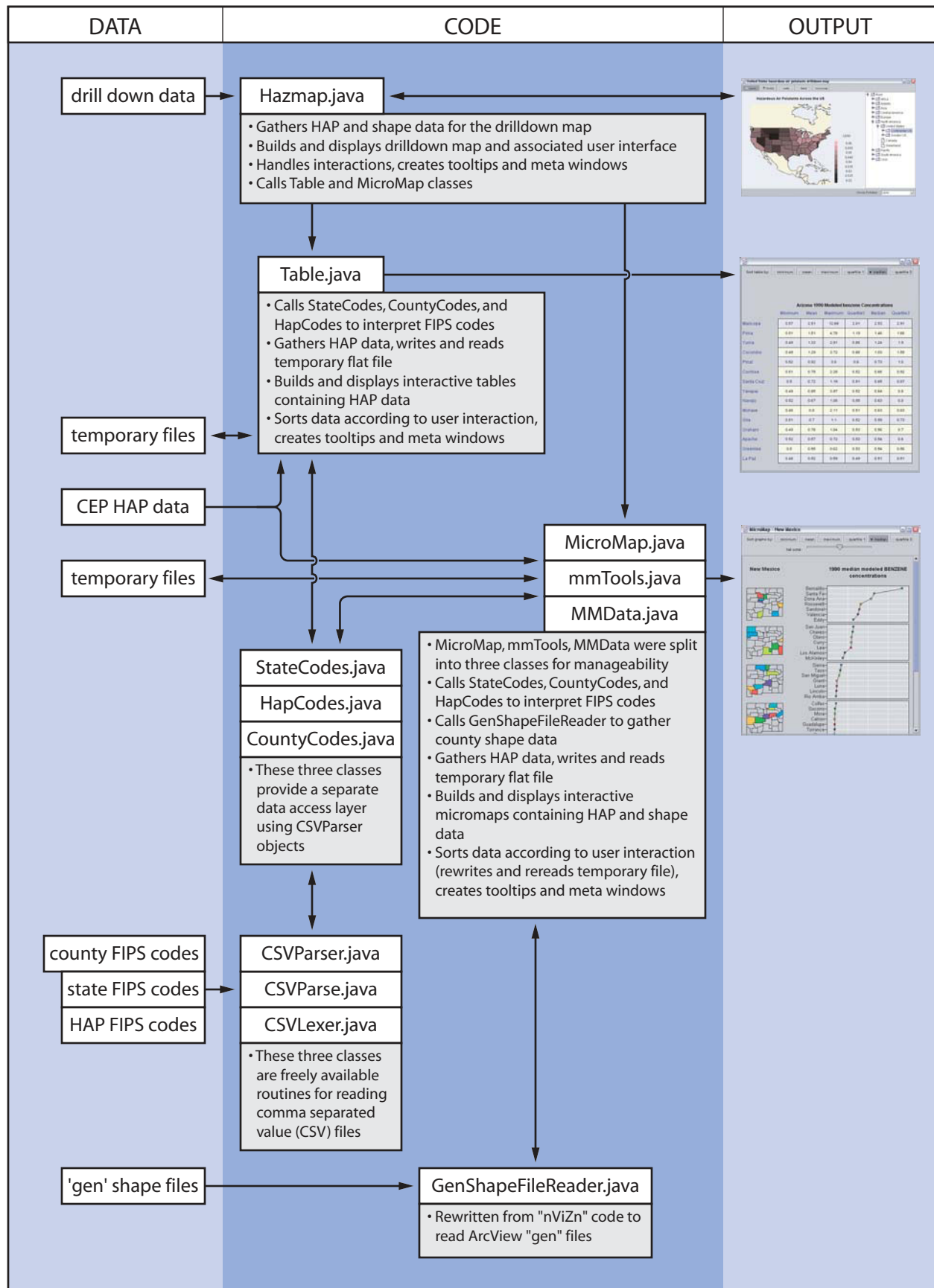


Figure 9—Software structure, data interactions, and output