CONSTRUCTING AND EVALUATING AN INTERACTIVE INTERFACE FOR VISUALIZING RELIABILITY

David Howard and Alan M. MacEachren
Department of Geography
302 Walker Building
Penn State University
University Park, PA 16802
Phone: 814-865-7600 (Howard)
814-865-7491 (MacEachren)
E-mail: DLH24@PSUVM.PSU.EDU (Howard)
NYB@PSUVM.PSU.EDU (MacEachren)

Abstract

Geographic visualization (GVIS) has become a basic tool of scientific research, especially research associated with the study of environmental change. Previous research into environments for interactive visualization has raised two questions [7]. These questions are:

1) what kind of user interface is most effective; and
2) what graphic variables are appropriate for showing different kinds of uncertainty.

This paper explores the first of these two questions.

One important issue for the study of environmental data is the reliability of the data. This paper describes the conceptual basis for and development of an interface for the exploration of reliability data. The model system is called RVIS for Reliability Visualization. RVIS was developed at The Pennsylvania State University to allow the visualization of environmental monitoring data and various representations of the reliability of that data. The intended users of the system include environmental scientists and policy makers. System evaluation follows a design suggested by Lindholm & Sarjakoski [5] that involves addressing interface design at three levels: conceptual, functional, and appearance.

1. Introduction

At a basic level, an interface can be defined as the 'place' where independent systems act on or communicate with each other. Computer information software such as a geographic information system (GIS) thus provides an 'interface' to systems in the world through the manipulation and analysis of data representing aspects of those systems. Within such software, however, the term 'user interface' is generally used to specify the parts of the program that provide the means for this connection or interchange. A user interface is the combination of program modules that query the user for input, interpret that input, and provide feedback to the user.

A good interface should be easy to learn and easy to use. It should facilitate simple, painless and intuitive use of a system. Lindholm and Sarjakoski say: "If the user thinks 'oh, what a nice interface', something is wrong. Ideally, the user should take no notice of the interface" [5]. Interfaces are built precisely so that people do not have to become experts in computer use and can spend their time being experts in their own discipline.

Jef Raskin [12], one of the developers of the Apple Macintosh, recently argued that even the acclaimed Macintosh graphical user interface gets in the way of what users really want to do (e.g., gain access to information on stocks, write a short story, etc.). He called for, and described the outline of, a more 'natural' interface that takes greater advantage of what we know about human cognition and behavior.

2. Interface Design Theory

Lindholm & Sarjakoski [5], following the example of Foley et al. [3], divide interfaces into three levels: conceptual, functional, and appearance. The conceptual level deals with the use and operation of the system in a general way. It involves consideration of the potential users of the system, their needs and training, and how best to meet these needs. The functional level of the interface encompasses the specific kinds of actions the user can perform and the meanings of these actions. At this level the specific capabilities of the system, and thus the controls that are necessary, are determined. The last level is the appearance level. Once you have decided what you want in the interface and how it will work, you must decide what the interface will (or should) look like. Now we will move on to a more detailed discussion of the three levels.

2.1 Conceptual Level

On the conceptual level, decisions center around the system as an interface to the data rather than the user's interface with the system. The needs of the user and characteristics of the data are the two things that must be uppermost in the system designers' minds at this level. When considering the needs of the user, Lindholm & Sarjakoski [5] present three questions to guide the interface builder: (1) what need does the system address, (2) how is this need met, and (3) what should be the result of working with the system. Of course, users are not a homogenous group. Different users have different needs. They might be interested in different implications or applications of the data and should be allowed to explore these varying interests. Perhaps the three things that would be highest on most users' wish lists for a data exploration system are ease of use, flexibility, and power. These three goals are often contradictory, but by keeping all of them in mind, designers can attempt to strike a balance between them.

2.2 Functional Level

When considering Lindholm and Sarjakoski's three conceptual questions, or more specific application-dependent goals such as those considered below in relation to RVIS, the interface builder starts to move into the functional level. The method by which actions are performed is one part of the functional level. Several things are important for the builder to think about in this regard. Consistency of function is among the most important. Similar tasks should be performed in similar ways. For instance, we would not want to have the user set data limits for one function by typing them in and set them for another function by moving a slider bar. Similarly, actions should not yield different responses in different contexts. Raskin [12] labels this interface problem a problem of modes and gives an example of a keyboard shortcut that sends the current document as an electronic mail message in one mode and that exits a document without saving in another mode -- an obviously undesirable situation. Feedback should also be consistent. This will lessen confusion and make the interface much easier to learn and use. There are several other considerations that should be kept in mind when designing each function of the system. For example, should an operation be performed interactively (as when selecting an area) or as a batch job (like printing) and are there some operations that should only be available some of the time or should all operations be available all of the time. There are often good reasons to create separate modes within which the interface is simplified by a restriction on commands that are available. Whenever modes are introduced, however, there is a danger of inconsistency or confusion as users move between them.

Some of the decisions at the functional level are very easy, but for others the alternatives do not have clearcut advantages. In choosing among alternatives the interface designer must also begin to consider the limitations of the hardware and software used.

2.3 Appearance Level

Some of these limitations may also constrain the appearance of the user interface and the clues to operation that the user will draw from it. This brings the designer to the appearance level.

Included here is consideration of layout (i.e. where to place controls and windows on the interface), as well as color schemes, legends, style of input, kinds of feedback provided and anything else that the user will have to see and decipher.

A large part of the appearance level involves choice of interaction style. In *Designing the User Interface*, Schneiderman [13] identifies five categories of interaction: menu selection, form fill-in, command language, natural language and direct manipulation. A short explanation of each follows.

A menu selection interface, (as the name implies), allows the user to interact with the system by selecting from menus or lists of possible actions. This approach reduces errors and helps remind the user what his/her options are at any point but also reduces the flexibility of the product by limiting options to a small number of predefined choices. An additional problem is that long menus, with multiple nestings, can be confusing for a user to peruse and the presence of multiple menus and submenus is often a hindrance to an experienced user.

Form fill-in presents the user with a set of fields to be filled in. This is perfect for querying or entering data into a data base but is not well suited to exploratory analysis, except perhaps in limited situations such as providing numerical bounds for data categories.

Command line interfaces involve keyboard input of commands, either interactively or to be stored for later use. This is a very powerful and flexible sort of interface but has several drawbacks. It requires that the user learn the commands and it generally can provide only limited protection against errors caused by mistyping commands or entering wrong command parameters.

Natural language interface commands are issued in the language that the user speaks and hears every day. As in command line interfaces commands are typed (although the ability to use spoken commands is becoming more realistic every day). Using natural language (rather than cryptic commands) makes the interface more comfortable for most users. However, natural languages, because of their ambiguities and nuances, are not well suited for computer comprehension. Thus such an interface must frequently ask for confirmation or clarification of commands, which quickly becomes tiresome. Also, as Lindholm and Sarjakoski point out, natural language is often already strained when trying to discuss spatial concepts which is why there are maps in the first place. A further issue, if the system will be used internationally, is that spatial concepts differ among languages [10].

The last style of interaction cited by Schneiderman [13] is direct manipulation. With direct manipulation interfaces the user, through a mouse or other pointing device, interacts directly with parts of the interface. Direct manipulation can involve either manipulation of controls on the display (through icons and tools) or manipulation of the display itself and objects in it (e.g., clicking on a map to zoom in on it). This interface style is the easiest to use (when well-constructed) and can be very efficient. It is also easy to learn and remember and may even stimulate exploration and experimentation. The direct manipulation interface is difficult to implement, however. Also, if it is not executed well, direct manipulation can be the most confusing of all of the types discussed either because users do not understand the purpose of some controls or because it is unclear which parts of the display may be manipulated.

In general, any specific tasks conceived of at the functional level can be presented through any of these interface styles. In fact, most interfaces that users are familiar with use several different styles for different operations. With today's visualization toolkits, we are not restricted to one style for all functions. An important job for the geographic visualization interface designer, therefore, is to choose the style for each operation that is easiest for the user to understand and use and then to mix these different styles with a minimum of confusion.

3. Construction of an Interface

To better explain the goals for our model project, we will begin by answering Lindholm and Sarjakoski's [5] conceptual level questions. To some extent, these are questions that we actually asked when conceptualizing our system. At that point, though, we were not following Lindholm & Sarjakoski's lead.

- **What need is (or should be) met by RVIS?** RVIS is intended to meet the U. S. Environmental Protection Agency's (EPA) need to examine spatial and temporal aspects of dissolved inorganic nitrogen (DIN) in the Chesapeake Bay. EPA also wanted to visualize the reliability (uncertainty) of the data as well as that due to data manipulation. Information about data reliability is a kind of metadata, data that is derived from and based on the original data or data surfaces. Specifically, in RVIS this usually means the reliability estimates given by the kriging procedure.
- **How is this goal (the goal of meeting the needs outlined in the first question) reached?** The goal can be reached by giving users a tool that allows data and data reliability to be examined independently, simultaneously, and as an integrated concept.
- **What should be the result of working with the application?** We hope that working with RVIS will result in more informed decision making in which analysts do not discount the possibility that the situation is worse (or better) than the best guess perspective (as shown by a single map display) might indicate.
- **Who are the users going to be?** (This question is not actually asked by Lindholm and Sarjakoski, but is suggested by Koussoulakou [4], and seems appropriate here.) The intended users are:
  1) Environmental Scientists
  2) Policy Makers

  The initial audience is the scientists, but policy makers are potential future users and we are trying to design RVIS to accommodate this audience as well.

After the basic conceptual level questions had been asked, attention turned to what functions RVIS should perform. Based on what RVIS is for and who the users might be, several specific functions were required:

- **The comparison of data surfaces from different times**
  The main method for doing this would be stepping through the data surfaces, in any order, at any speed. This would allow the user to compare any two data surfaces and to investigate process by viewing time slices as an animation. Since there might be seasonal or annual cycles, we also wanted the user to be able to choose any data surface by date and compare or aggregate information for the same point in the cycle across years.
- **Representation of reliability**
  At this point, we have estimated two different kinds of reliability for this data set. The first is reliability at the data collection sites which is determined by 'jacknifing'. *Jackknifing* is a technique that estimates what would happen if individual data values were missing from the interpolation process. The second kind of reliability emphasizes the uncertainty due to areal interpolation from a relatively sparse set of 49 sample points. In this case, we are representing the confidence limits for each grid cell given by kriging. These confidence limits are a measure of how likely a given point obtained by interpolation is to lie within some interval of the estimated value. Representation of both point and area reliability is important. An additional goal is to assess and represent temporal reliability associated with factors of data collection (such as the number of days it took to collect each sample). Finally, we plan to represent combined temporal-spatial reliability.
- **Comparison of data and metadata**
  Three methods of comparison are desirable. The first, and the default, is side-by-side maps. The second is merging of the data and metadata. This is done both by overlaying the data with

the metadata (using symbolization that keeps each component visually separate) and by merging the two into one bivariate map. The last method is alternation between the data and metadata, both as an animation (resulting in a 'flickering' display) and as user-controlled toggling back and forth.

- **Focusing**
  Focusing is a term borrowed from the exploratory data analysis (EDA) literature. It refers to the ability to set display thresholds that will restrict the part of the data being viewed. The most important kind of focusing in a system like RVIS is to isolate those values above or below a threshold. This is applied to both data and metadata.

A key problem faced in the design of RVIS was to match the functional goals for the system with the tools that can be built with Interactive Data Language (IDL). IDL allows a designer to build an interface and guide a program via tools called 'widgets'. A widget is a simple graphical object, such as a pushbutton, slider, or menu that allows users easy interaction with the program. Interaction with a widget produces what is referred to as an 'event' from that widget. When the user generates an event (by pushing a button, moving a slider, or otherwise manipulating a widget), the software is then able to respond to the event by performing some function.

Six different types of widget are available in IDL:

Non-manipulable widgets

- Base widgets: that allow the designer to specify a base (including its position on the screen, color, etc.) upon which other widgets will appear.
- Drawing widgets: these allow the user to display graphics. They must be rectangular but can be of any size.

Manipulable widgets

- Slider widgets: in which a user can move a control back and forth along a sliding scale.
- Button widgets: that can be labelled to correspond to some action which will be taken if a user clicks on them.
- Text widgets: that provide for the entry or display of text (or numerical information).
- List widgets: these consist of either pull-down or pop-up menus that allow choices of actions from a menu.

These widgets allow users to interact with a system built in IDL using several of the interaction types defined by Schneiderman [13]. Menu selection, form fill-in, and direct manipulation of controls are all supported by IDL widgets. Limited direct manipulation of the display is also possible.

Our user interface went through many revisions as new modules were added to the system and continues to be revised. At first, we started at the top of the list just presented and began to match widgets to functions and then to make these widgets generate the correct events. Thus the first widgets on the panel were a drawing window used to display the data surface and a sliding bar that allowed the user to 'page' through the surfaces in chronological order. Controls and displays were then added as necessary. A picture of the present version of RVIS is provided in Figure 1.

The present interface to RVIS contains two map windows with a central panel of controls. The two map windows allow side-by-side comparison of any two maps. The controls all activate one or more of the functions that we listed earlier. For instance, the slider bar labelled 'graymax' controls a function that allows the user to 'focus' on a section of the reliability map. As the slider is moved, a

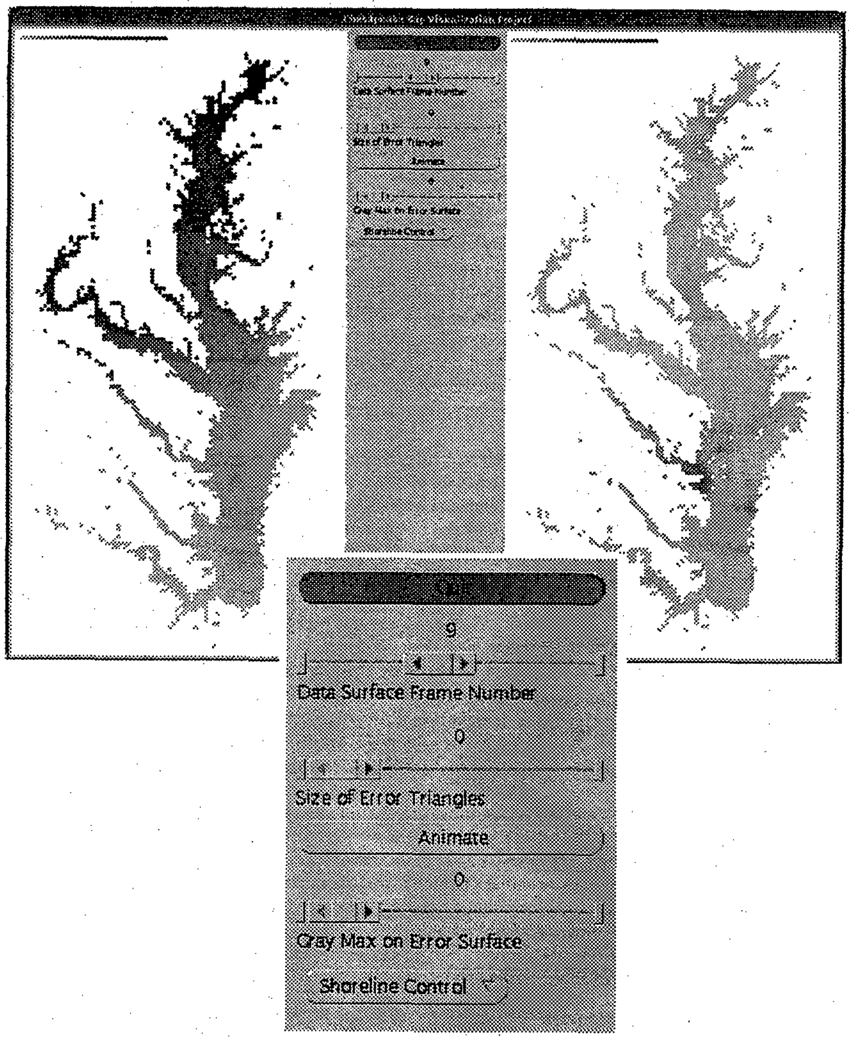

Figure 1. A sample screen from RVIS (above). The actual display is in color and viewed on a 21 inch monitor. The map on the left displays dissolved inorganic nitrogen with darker representing more nitrogen (using shades of red). The map on the right represents reliability (measured as the variance estimate from kriging). The original uses shades of blue, with darker representing higher levels of uncertainty. The contol panel from the sample screen is enlarged (below). It illustrates both button and direct manipulation slider widgets.

reliability threshold is increased and the more highly reliable areas are turned to gray. The result is an emphasis on areas of lower reliability.

Color schemes were given particular attention as we implemented RVIS. IDL allows colors to be defined in any of several ways including red/green/blue and hue/value/saturation. Because we wanted to discuss various display schemes in terms we, and most cartographers, were familiar with, we adopted the hue/saturation/value system. Brewer's [1] color syntactics for univariate and bivariate mapping played a key role in designing color schemes to match functional goals.

Based on MacEachren's [7] earlier proposals about uncertainty representation we planned to experiment with color saturation as a way to depict uncertainty. A logical match for saturation-uncertainty is to apply color value differences to the data. Therefore, initially we tried a continuous value scale in one hue for the data and a saturation scale for the metadata. However, these schemes did not provide enough contrast for the data sets we tried, especially when printed on paper. Identical value/saturation scales using different hues for data and metadata proved to be more successful at highlighting extremes of data and/or metadata. When applied to side-by-side map pairs, this dual-hue depiction makes interpretation easy for the user because the schemes do not require switching 'schemata' when looking from one map to another (in other words, once the user has seen that 'dark = more' and 'light = less' on the data map, he/she will not have to switch this interpretation when looking at the metadata surface).

Intuition suggests that high DIN values, being those that exceed tolerances by the largest margin, should be highlighted, and thus represented by the most noticeable color. Given the light-grey background that we chose, we decided to follow the normal convention for paper maps and depict high DIN values in what we hope is a menacing dark red. Lower values are shown in lighter colors. We chose a light grey background because a) the extremes of black or white for backgrounds were harsh and overwhelming and b) because McGranaghan [9] found that using dark colors on a light background lowers the chance that the map reader will become confused about the order of the data.

Another consideration when designing the maps displayed by RVIS was map scale. There were many constraints here, including screen size, desired resolution, and the size of array needed to define the map. There is a trade-off between the desire by analysts for spatial precision in the maps and the resolution at which the computer system can support interactive analysis. In addition, small cells imply higher interpolation accuracy than the sparse 49 point sample network can achieve. We settled on 25 screen pixels per cell, which gave us a 90 by 72 array of four km^2 cells. These cells can be seen individually on the display, yet are not so distinct as to produce a disjointed or blocky look. The resolution is only one-half that used by the EPA in previous analysis of the same data. A sample network of 49 sites to cover Chesapeake Bay, however, did not seem to justify interpolation to the one km^2 cells they have used.

Within RVIS, we have implemented three basic styles of presenting data and metadata. These are side-by-side display, bivariate maps (which include both overlaid symbols and merged symbols) and alternation. Side-by-side is what we chose as our initial default display. Though people do not do a very good job of comparing maps, as demonstrated by Lloyd and Steinke [6], this display has the distinct advantage that the data and metadata maps can initially be looked at separately. We can generally assume that most analysts will want to consider the data independently before taking data reliability into account.

After bringing up the initial side-by-side display, the analyst can select among various options for overlay and merging of maps. We implement the overlay technique in two ways, both of which borrow from ideas presented by DiBiase, et al. [2] for display of multivariate information. One variable (the data) is depicted with area shading and the second (the metadata) is depicted with line or point symbols. One of our overlay displays uses weighted isolines to depict the metadata, where the isolines get thicker as the values get higher. The second kind of overlay uses scalable triangles representing the potential errors at sample points when/if those sample points are missing. Both of these are tools that can be turned on and off within RVIS.

Overlay methods emphasize the data while allowing analysts to check the quality for map areas that seem to be particularly good or bad in terms of meeting the EPA dissolved inorganic nitrogen targets. The merged display style uses bivariate maps to put emphasis on relationships between data and reliability. For example, a bivariate map can draw attention to the fact that some areas that appear to be meeting standards also have low data reliability and thus could be exceeding those standards.

The bivariate maps in RVIS were created by grouping both the interpolated data and interpolated metadata into classes. Both the data and metadata were broken into three equal-interval classes. We chose to use three classes for each variable based on conclusions drawn by Olson [11] while studying bi-variate Census maps. Many coloring schemes, most suggested by Brewer's [1] research into color choropleth mapping schemes, were implemented for the 3-x-3 bivariate maps. All of these representations are available by toggling the bivariate map on and choosing among the coloring schemes from a menu.

One additional bivariate scheme that is implemented is not a traditional representation. This scheme can be thought of as a reliability filter. In this representation, the map cells in the lowest class of reliability have all but one pixel turned grey so that the data value is obscured. In the medium reliability class, 16 of the 25 pixels in each cell are desaturated. In the highest class, all pixels are left at their original color. This mapping scheme focusses the user's attention on the areas of the map that are highly reliable while not totally hiding the less reliable data values.

One last way for the user to easily compare the data and reliability surfaces is through alternation. Alternation is achieved in two ways. In the first method, the user simply displays a data map and then the corresponding metadata map in succession. By toggling back and forth between them, the user is able to alternate the two surfaces in a self-paced manner. The second method of alternation is an animation that takes the currently selected pair of data/metadata surfaces and alternates between them several times (a technique that has also been called flickering).

4. Summary/Conclusion

Dynamic mapping tools have become increasingly accessible to those for whom a spatial perspective is important. Scientists and policy analysts are no longer content to wait for cartographers to produce maps - maps are being produced as the need arises by those who need them. Although it would be difficult to verify, it is probably safe to contend that the number of unique "virtual" (or softcopy) maps produced each week far exceeds the number of paper map titles. Science is being done and policy decisions are being made on the basis of virtual maps. Although some of these maps go on to achieve a tangible status in paper or electronic (e.g., CD-ROM) form, the role of these maps in thinking and policy formulation is often over before this transformation happens.

As a result of the above, Geographic Visualization is a critical research and application area for cartography as we approach a new century. As defined by MacEachren [7], GVIS is typified by map use in which *individuals* are actively engaged in data *exploration* (searching for unknowns) using tools that are highly *interactive*. This characterization of GVIS points to several gaps in knowledge and method related to display and use of geo-referenced information. If cartographers do not address these gaps in knowledge, someone else will or, perhaps worse, no one will. The areas requiring research include a range of conceptual and technical issues related to (a) design and production of dynamically manipulable maps (and related geographic representations) and (b) the use of maps in scientific research (e.g., to generate hypotheses) and policy analysis (to support decision-making). These research areas are linked through a set of common issues related to development of user interfaces to geo-referenced information.

Although cartography as a discipline has centuries of experience with design of maps as an 'interface' to geo-referenced information, we have limited experience with design of interactive computer interfaces or with *design of maps as an integral component of such interfaces*. There is, of course, much to learn from the interface design literature in computer science and human factors engineering -- but geo-referenced information has different characteristics that require specialized handling. There is a need, and an opportunity, for cartographers to have a significant impact on science and policy desicions through effective interface design for GVIS (and for GIS). In addition, cartographic innovation in GVIS interface design is likely to have an impact beyond GVIS in other areas of interface design -- assuming we make the effort to communicate our research beyond the bounds of the discipline.

The project discussed above is an attempt to extract some key principles from the general field of computer interface design (along with some techniques borrowed from exploratory data analysis) and adapt/extend them to a GVIS context. We have approached this task largely through the vehicle of a case study that emphasizes GVIS tools for analysis of environmental change. Within this application area, the issue of representing data relibaility has been identified as a critical concern. The resulting GVIS environment is still in prototype stage. Goals for the future are to expand upon the systems' capabilities, explore the role of metaphor as a mechanism for building logical structures into the system, and to conduct an empirical evaluation of the systems' use by environmental scientists and policy makers.

5. References

[1] Brewer, C. 1994. Color Use Guidelines for Mapping and Visualization. In *Visualization in Modern Cartography*, ed. A. M. MacEachren and D. R. F. Taylor. London: Elsevier, 123-147.

[2] DiBiase, D., Reeves, C., MacEachren, A. M., von Wyss, M., Sloan, J. L., and Detweiler, M. C. 1994. Multivariate Display of Geographic Data: Applications in Earth System Science. In *Visualization in Modern Cartography*, ed. A. M. MacEachren and D. R. F. Taylor. London: Elsevier, pp. 287-312.

[3] Foley, J. D., van Dam, A., Feiner, S. K., and Hughes, J. F. 1990. *Computer Graphics: Principles and Proactice*. Reading, Mass.: Addison-Wesley.

[4] Koussoulakou, A. 1994. Spatial-temporal Analysis of Urban Air Pollution. In *Visualization in Modern Cartography*, ed. A. M. MacEachren and D. R. F. Taylor. London: Elsevier, pp. 243-267.

[5] Lindholm, M. & T. Sarjakoski. 1994. Designing a Visualization Interface. In *Visualization in Modern Cartography*, ed. A. M. MacEachren and D. R. F. Taylor, London: Pergamon, pp. 167-184.

[6] Lloyd, R. E. & T. R. Steinke. 1976. The Decisionmaking Process for Judging the Similarity of Choropleth Maps. *The American Cartographer* **2**:177-184.

[7] MacEachren, A. M. 1992. Visualizing Uncertain Information. *Cartographic Perspectives* **13**: 10-19.

[8] MacEachren, A. M. 1994. Visualization in Modern Cartography: Setting the Agenda. In *Visualization in Modern Cartography*, ed. A. M. MacEachren and D. R. F. Taylor, London: Pergamon, pp. 1-12.

[9] McGranaghan, M. 1989. Ordering choropleth map symbols: The effect of background. *American Cartographer* **16**(4): 279-285.

[10] Mark, D. M. and Egenhofer, M. J. 1995. Topology of Prototypical Spatial Relations Between Lines and Regions in English and Spanish. In *Auto-Carto 12*, ACSM/ASPRS Annual Convention, Charlotte, NC, February 27-March 2, 1995, pp. 245-254.

[11] Olson, J. 1981. Spectrally-Encoded Two-Variable Maps. *Annals of the Association of American Geographers* 71(2):259-276.

[12] Raskin, Jef. 1995. The Interface Paradox. Presentation for Penn State Departments of Computer Science and Engineering, April 11, 1995.

[13] Schneiderman, B. 1987. *Designing the User Interface*. Reading, Mass.: Addison-Wesley.