

GeoJabber: Enabling Geo-Collaborative Visual Analysis

Frank Hardisty

GeoVISTA Center, Dutton e-Education Institute

Department of Geography

The Pennsylvania State University

Email: hardisty@psu.edu

Surface address: 212 Walker Building, University Park, PA, 16802

Telephone: 814-867-1471

Fax: 814-863-1564

GeoJabber: Enabling Geo-Collaborative Visual Analysis

Frank Hardisty

GeoVISTA Center, Dutton e-Education Institute Department of Geography, The Pennsylvania State University

Abstract

The GeoViz Toolkit introduced here is an open source software project that enables multivariate visual analysis of geospatial data. The open XMPP communication protocol (also known as Jabber) was used in the GeoViz Toolkit software to create a working prototype of a geocollaboration system, by creating extensions to Jabber to support tool state sharing, including geospatial aspects of tool state. Advantages and disadvantages of using XMPP vs. other implementation methods are detailed.

KEYWORDS: Collaborative and Distributed Visualization, Human-Computer Interaction, Geographic Visualization, High-dimensional Data, Visualization system architectures, toolkits, and problem-solving environments

Introduction

A recent report by the National Research Council asserts that “enabling collaborative work with geospatial information” is a key research challenge to support in activities in many domains, including homeland security (NRC 2003). A scenario that illustrates this would be that in the event of a terrorist attack, multiple people would need to coordinate their response to the attack at the same time. A map interface that supported input from multiple users could be critically important. The unfortunate truth is that current geospatial information technology does not support collaboration between users in different places working on the same problem at the

same time. This paper presents a test system that would investigate how to mediate between users who are working on the same geospatial problem in different places.

The core problem that this research attacks is “how can users’ actions in geovisualization software in different places be coordinated?” The overall approach taken here is to rely on marshalling and unmarshalling of software objects into XML representations, which can then be inserted into Jabber (XMPP) packets and shared between users. In the remainder of this paper, first the GeoViz Toolkit is introduced, next a typology of what is to be coordinated between users during a collaborative analysis session is introduced, and finally the GeoJabber implementation is presented, with other implementation alternatives described.

The GeoViz Toolkit

The GeoViz Toolkit is an Open Source project which enables multivariate exploration of geospatially referenced data sets. Its architecture is that of a set of software components which become connected at runtime. This lends itself well to coordination across the network, because a remote sender of information can be modeled as another component in the framework. The built-in components include tools for data handling, univariate, bivariate, and multivariate visualization, multiform matrix visualization, dimensional reduction techniques, animation and conditioning, and some spatial analysis methods, as well as a GeoJabber support. One configuration of its user interface can be seen in Figure 1 below.

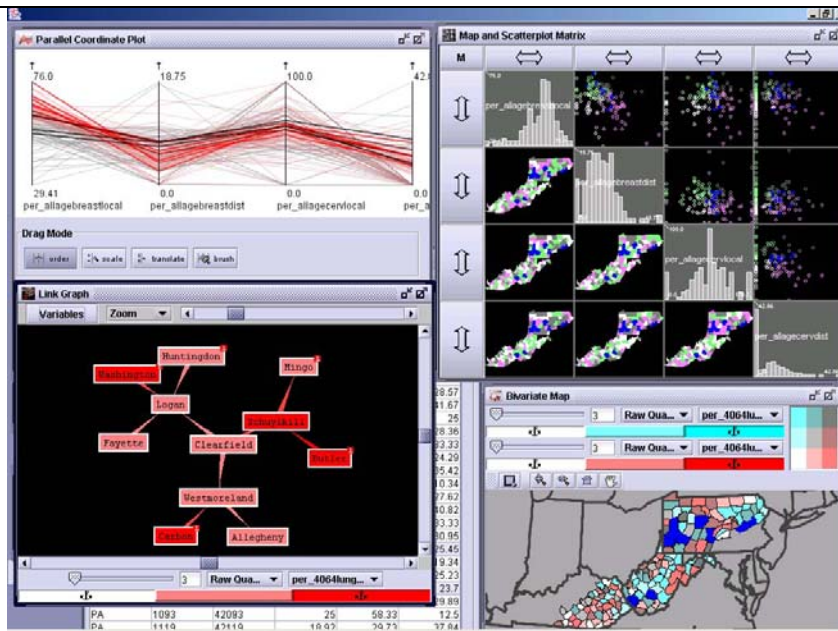


Figure : GeoViz Toolkit Interface

Example GeoJabber Workflow

Categories of Coordinated Visual and Numerical Aspects of Data Representation

In this section, we take a more detailed look at what coordination is to be done, as opposed to how to do it. All of these event types are or will be supported by GeoJabber. We do this by first examining previous work in the area, then by going into some more detail about the mechanisms used to implement coordination in the current research, events and event listeners. Additionally, we will examine the rationale and data structures used for different categories of coordinated visual and numerical aspects of data representation.

Identifying the categories of coordination among types of visual and numerical aspects of data representation is a critical task for geographic visualization software for enumerated data. It is related to, but separate from, the task of identifying GIS operations (Albrecht and Kempainen 1996; Albrecht 1999), space-time operators (Qian et al. 1997), and interactivity types in geovisualization (Crampton 2002). Albrecht provides a set of universal GIS

operations, including the following categories of operations: Search, Location Analysis, Terrain Analysis, Distribution/Neighborhood, Spatial Analysis, and Measurements. Crampton offers the following four categories of interaction: (1) with the Data; (2) with the Data Representation; (3) with the Temporal Dimension; and (4) Contextualizing Interaction.

Albrecht's and Crampton's categories, while helpful, are in some aspects orthogonal to the categories of coordination being developed here. For example, coordination itself is considered a Contextualizing Interaction in Crampton's typology, as is having multiple views on the data. Similarly, most of the operations in Albrecht's work are specific to the spatial aspect of the data, aside from the "search" operation, which maps to selection. One of the conceptual bases for Crampton's typology is a single, integrated map visualization component to which other components may be linked. The GeoViz toolkit was developed with the concept of co-equal components, none of which has a central role, except perhaps the coordinator itself. I draw on Crampton's first three categories when defining my own (see below).

More open-ended, and therefore more applicable to the task of defining what categories of coordinated geovisualization are appropriate, is a taxonomy of visualization goals, presented in (Keller and Keller 1992). These "goals" are categories of visualization strategies, identified as pairs of two categories, an "action" and "data". Types of action include: identify, locate, distinguish, categorize, cluster, rank, compare, associate, and correlate. Types of data include: scalar, nominal, direction, shape, position, spatially extended region, and structure. Thus, highlighting selected observations in a scatterplot would fall under the goal "identify cluster", highlighting in a map would be "identify spatially extended region". What is needed for categories of coordination is closer to the "action" categories, since different visualization components will have different "data" types that they operate upon, for example, the scatterplot and map operate on different sets of data, but may share "action" types.

Keim (Keim 2002) provides an interesting classification of visual data mining techniques that also has relevance here. He identifies data types (one-dimensional, two-dimensional, multidimensional, text, hierarchies, algorithms), visualization techniques (2D/3D plots, transformed displays, icon-based displays, dense pixel displays, and stacked displays), and interaction techniques (interactive projection, interactive filtering, interactive

zooming, interactive distortion, interactive linking and brushing). As candidates for types of coordination, the data types and the interaction techniques are both possibilities, and representatives from both are present in the categorization presented here.

The following are proposed as primary types of coordinated visual and numerical aspects of data representation: **data**, **display**, and **category**. These are arranged in order of likely dependence in the construction of a geovisualization view, which can be shared via GeoJabber. *Data* coordination is the coordination of the set of entities under analysis. Data comes first, because it is the “universe” that all other operations are applied to. Examples of data coordination would be applying the same overall data set to a number of components simultaneously, and extending the data set to include a derived field for each entity. *Display* coordination is the coordination of representation methods. Examples of display coordination would be using the same data-to-display size mappings in multiple components, and applying the same background color in multiple components. Displays follow Data because the user may often wish to vary the symbolization to better explore the data. *Category* coordination is the coordination of divisions (or groupings) in the data. Examples of coordinated categories include linked brushing between components (where categories are “highlighted” and “not highlighted”), and focusing on the same data range in different components. Categories come last because they encompass such transitory operations as which observation is being currently examined by the user. Each of these types represents fundamental operations in geographic visualization that apply to many kinds of visualization components. Each of these may also be expanded into subtypes. Below, the types and some sub-types are expanded upon, and then the current set of events supported in the GeoViz Toolkit and GeoJabber is mapped onto these types.

Data

The “data” type includes events that carry the information that there is a new or different set of data to analyze. This kind of event indicates that the data space being analyzed has been changed. For example, if a spatial data set representing the provinces of Nepal replaces a data set consisting of the states of the United States, this should be communicated to any coordinated components and coordinated users. Extensions to the original data

observations, including calculated fields or data linkages, would also be communicated using data type events. Similarly, extensions to variables, such as metadata on the origin and accuracy of different variables, would be communicated with data events.

Display

If the user has assigned some visual representation to some observations, these should be widely communicated and used. This coordination can enable discovery of spatial patterns based on non-spatial attribute data, and exploration of particular places in attribute data. Symbolization events could include information about many kinds of data to display mappings. Subtypes of symbolization include static visual properties, and dynamic visual properties.

Static visual properties useful for representing data identified by MacEachren (MacEachren 1995) include: location, size, crispness, resolution, transparency, color value, color saturation, color hue, texture, orientation, arrangement, and shape. A similar set was parsed by Wilkinson (Wilkinson 1999), following Bertin (Bertin 1981), into Form (size, shape, rotation), Color (hue, brightness, saturation), Texture (granularity, pattern, orientation), and Optics (blur, transparency). These subtypes (form, color, texture, optics) are a promising avenue to explore for coordinated symbolization.

Dynamic properties include variables applying to whole scenes, and to individual observations. Three "dynamic variables" – scene duration, rate of change between scenes, and scene order – were initially identified by DiBiase et al. (DiBiase et al. 1992), to which three more were later added display date, frequency, and synchronization (MacEachren 1995). All of these can also be applied to individual observations as well, as could jitter and “motion paths” (Ware 2000).

Other important sensory modalities of information representation include sound (Krygier 1994) and touch (Griffin 2000). Further, data-to-display mappings and coordination of display form (e.g. a change from a choropleth map to a graduated circle map that should be reflected in all maps in a matrix) should be included here.

Category

Category type operations include many types of sub-setting operations on data sets. These are subdivided into “extent” types of coordination, and “classifying” types of coordination. The extent type includes attribute, spatial, and temporal, as further sub-types. The attribute subtype includes user-driven selections, as well as focusing and indication operations. Classifying includes such potentially coordinated aspects as traditional cartographic classifications, for example, selecting among quantile, equal interval, and minimum variation (optimal) classifications, and more complicated multivariate classifying, such as K-means clustering. The simple act, by the user, of selecting some observations as being of particular interest is a powerful indicator that they are important to the analysis. Thus, it is essential that classification events be supported by GeoJabber.

Attribute Extent — Subsetting

Subset coordination is the most common type of coordination among other coordination approaches; see (North and Shneiderman 2000) and (Hurley 2000), and has been used for decades in statistical graphics (Fisherkeller, Friedman, and Tukey 1974; Becker and Cleveland 1987; Cleveland 1987). Subsetting remains one of the most important types of coordinated event in a linked toolset. A user created subset is, almost by definition, a way that the user has of declaring, “these observations are of special interest”. Subsetting can be achieved through several types of action, such as brushing, indication, focusing, and conditioning. All of these action types produce subsets from a total collection of observations; in other words, they could be expressed using SQL “Select” statements.

Subspace changes, which are another type of subsetting, indicate that there is a different set of variables to analyze, consisting of some subset of the whole, for those analysis components capable of visually or numerically analyzing multiple variables. For example, if our whole data set had fifty variables, but we wished to examine six in more detail, this would be communicated as a subspace change. Subspace changes can also communicate the order that the variables should be presented in. This order is important because proximity of variables in tools like the PCP or matrix helps the analyst extract interesting relationships. One of the tools in the GeoViz toolkit, the Subspace Linkgraph, provides just such functionality.

Brushing is subsetting driven by user interaction, as in “linking and brushing”. An *indication* is a transient subsetting (Raskin 2000), normally of just one observation. Indication events are normally triggered in a graphical user interface by a user pausing her pointer over a particular observation. This observation can then be separately

highlighted in multiple representation components. Excentric labeling (Fekete and Plaisant 1999) provides multi-observation indication. Excentric labeling is described and illustrated below, in reference to Figure 4. *Focusing* signifies that a subset of the original data is being emphasized (either by entirely removing objects that are not within the “focus” of view or by otherwise changing appearance of the focus and/or non-focus objects). This differs from brushing in that brushing highlights while focusing hides. *Conditioning* is the complement of focusing, but applied to a variable other than the one displayed. Conditioning, in the visual sense, consists of filtering the data set by some variable, and excluding from visibility those observations that lie outside the range for the conditioning variable. Conditioning can be used to explore relationships between correlated variables, seeing the remaining distribution in one variable when observations with a given data range in the other variable are filtered out (Carr, Wallin, and Carr 2000).

Spatial and Temporal Extent

Spatial, as a category, includes operations on the geographically referenced portion of a data set. These include panning and zooming operations, changing the field of view for the user. Both raster and vector types of spatial data operations need to be represented. Temporal extent would change the range of times that form the current view of the phenomenon under analysis. The user should be able to choose whether or not a temporal animation in one view coordinates with temporal animation in other views and if it does whether they are synchronized in time or offset in time.

Classifying

Classifying consists of categorizing the data based on some (often numerical) aspect of the data. It is a fundamental technique in geographic visualization (Mak and Coulson 1991; Egbert and Slocum 1992), in machine learning techniques (Whigham, McKay, and Davis 1992), and many other related fields. The *Journal of Classification* lists the following fields from which it solicits contributions:

statistics, psychology, biology, information retrieval, anthropology, archeology, astronomy, business, chemistry, computer science, economics, engineering, geography, geology, linguistics, marketing, mathematics, medicine, political science, psychiatry, sociology, and soil science (Heiser 2003)

Classifying is thus a critical aspect of geographic visualization to support across different components. One can, for example, classify the data using a self-organizing map technique, and see the results in a parallel coordinate

plot and a choropleth map. The users' choices for classifying are critically important information that should be leveraged by every component that can do so. In other words, if the user has arranged for the data to be grouped together on some basis, these groups should be widely communicated and used. The groups are the best guess that any component could have about what divisions in the data to highlight.

Summary of Data Representation Typology

The framework outlined above characterizes the kinds of coordination that can be implemented in a multi-component, multi-view geovisualization environment (or InfoVis environment more generally), dividing kinds of coordination into data, display, and category types. The coordination mechanism in GeoJabber ensures that both the set of coordinated aspects of visual and numerical representation supported by any client and the total set of coordinated aspects can be added to incrementally. In other words, for each component, the number of coordinated aspects can be expanded over time, and their organization may be modified. At first, a component may only support common subsetting, but may add support for a common coloring scheme later, without breaking any kind of compatibility. Additionally, new coordinated aspects can be added as needed. So, if "fill order" was going to be coordinated, it could be added without interfering with the existing set of coordinated aspects of geovisualization component behaviour.

Events: A mechanism for Coordinating Inter-Component Visual and Numerical Aspects of Data Representation

All of the coordination that takes place here uses the mechanism of software "events". Software events, like their real-world counterparts, are "things that happen". Events in Java were originally used exclusively for communication between graphical user interface (GUI) components. For example, a slider with a drag handle and a text box could communicate using events. Every time the user dragged the slider, the slider would fire an event, and the text box could receive it and update its internal state and appearance accordingly. However, recent versions of Java include a more generic event class, which supports communication between any components. Microsoft's .Net platform includes similar event support, as do libraries for C++, Python, PHP, and many other languages. Event-based programming has been promoted as an aid to creating robust, scalable computer programs

(Pietzuch and Bacon 2002). All information in these events can be transmitted between components, for example inside the GeoViz Toolkit, or across the network, when using GeoJabber.

An important counterpart to the event is the event listener. An event listener is a class that implements an interface guaranteeing that the listening class will receive the event. With this bit of indirection, the concept of events and event listeners becomes much more flexible. Any class that wants to receive events from another class can register with the sending class. The sending class then need only keep a list of classes that have registered, and send events to these classes whenever the event trigger happens. Taking our previous example, a text box would have to implement the “change listener” interface to receive “change events” from the slider. Conversely, the slider could register with the text box to receive notification when the contents of the text box have changed.

All of the events used in the coordination system presented here are subclasses of the EventObject class. This class only has one method of interest, called getSource. This provides a reference to whatever class sends the event. Normally, events will provide access to additional methods that, in turn, allow access to some important data structures, either directly or as a call back to the sending class. For example, if a selection is changed in one class, then events will be broadcast that allow the receiving classes to find out what the current selection is.

What follows in the remainder of this section is a description of types of events that are coordinated in the GeoJabber (and the GeoViz Toolkit). These are grouped into three main groups, as described above, data type events, display type events, and classifying type events. These follow the path from the most stable to the least stable, in terms of how often these events are fired in GeoJabber. The events that are fired the least often represent the most stability from the application’s standpoint.

Data type events

In the GeoJabber, the only data type event that is currently implemented is the dataset event but more of these events will be supported in the future, such as data extension events. Dataset events are supported by almost every component, because the spatial and tabular data that are being visualized are carried in a dataset event. A dataset event carries a reference to a class, called DataSetForApps, which carries the data as a set of arrays.

Display type events

Symbolization can take many forms. The visual aspect component of visualization that is best supported in the current set of coordinated events is assignment of *color*. Two types of coordinated color events have been developed, one that was intended for components to straightforwardly assign a color to each observation, and another that encapsulated the information necessary to start from raw data and derive a classification and assign colors to the resulting classes.

The first approach, which is more passive on the part of those components receiving the colors, is appropriate for components that do not assign color themselves. For example, a parallel coordinate plot could receive this type of color event and render each observation (thus each string) in the specified color. This coordination mechanism does not require any knowledge of variables displayed in the PCP; observations can be assigned a color based on any criteria (e.g., membership in a category for a variable not displayed). The data structure attached to the event is an array of colors, which has a length equal to the number of observations in the data set. The receiving component then merely needs to apply the colors to the observations when rendering, as illustrated in Figure 2.

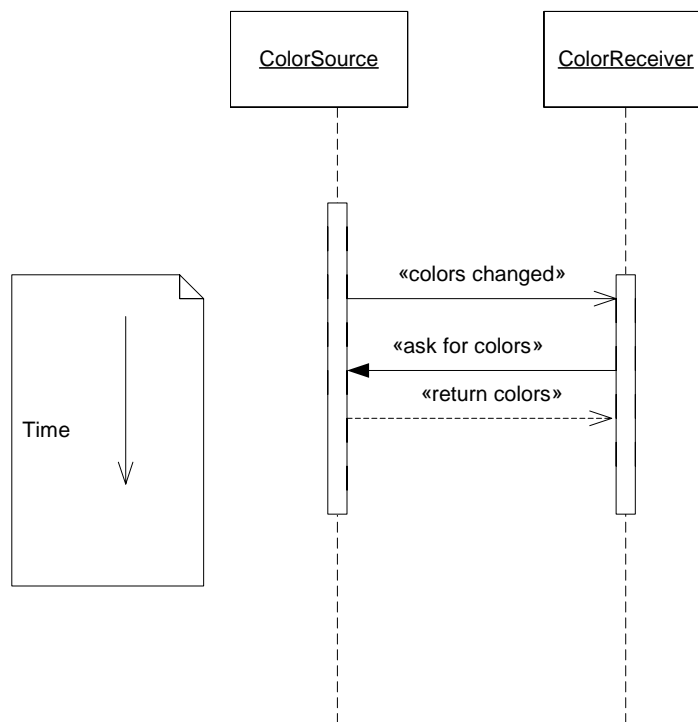


Figure : UML sequence diagram of the array of colors approach to passing color information

The second, more encapsulated approach is appropriate for those components that base their coloring on their position in another component, as in the matrix and small-multiple components associated with the GeoJabber. The data structure passed is actually a class that can perform classification and coloring on the data, called in our system a ColorClassifier. The receiving component then passes data to this ColorClassifier and receives back a set of colors to apply to the observations. Figure 3 illustrates some of the program flow involved in implementing this method.

The matrix component needs this kind of color information encapsulated in a display event because the matrix passes the symbolization information to its various elements without knowing what the coloring of each matrix element should be. One of the key capabilities of the matrix is that it does not impose symbolization, but enables each element to symbolize itself based on position in the matrix. For this kind of information to be passed from one component to another, some form of abstraction like the one shown in Figure 3 is necessary.

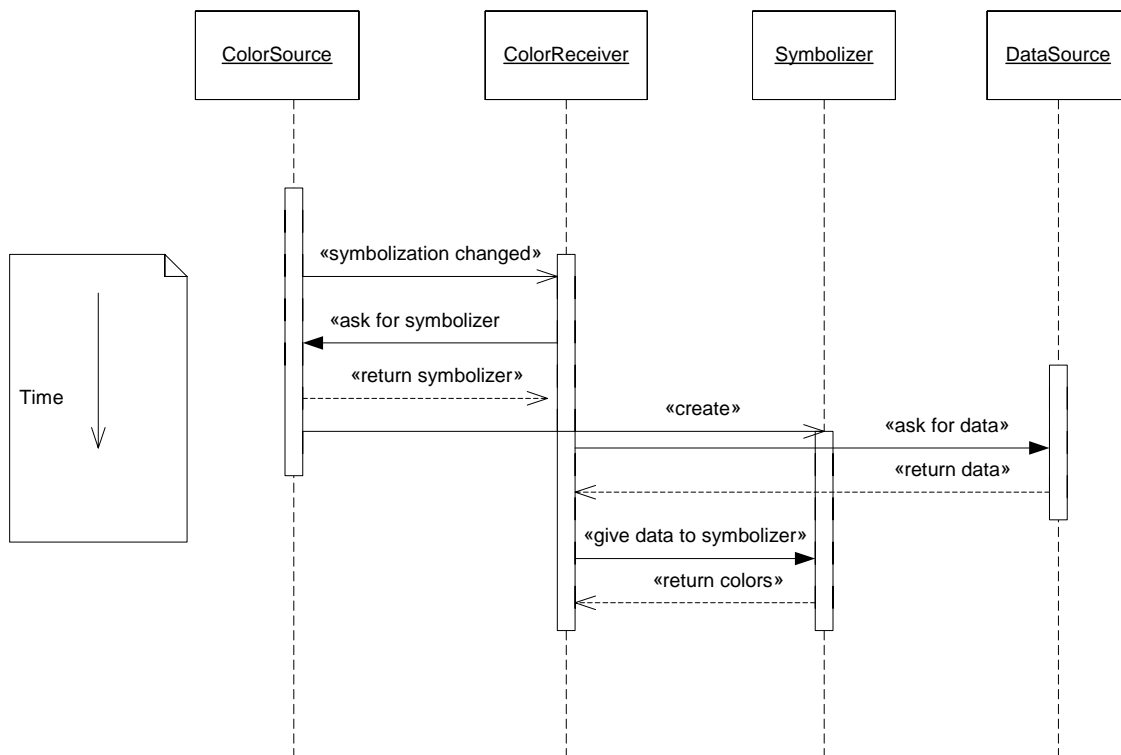


Figure : UML sequence diagram of reference to symbolizer approach to passing color information

Colors are tied to classifications in this scheme, but the coloring and the classification are separate objects. Thus, if future extenders of these components wish to combine variables like shape or texture with classifications, the extension mechanisms should enable the reuse of the classification objects.

Category events

Category events include ones to define extents as well as events defining coordinated changes in classifying. The extent types instantiated include attribute and spatial events. The classifying events implemented include conditioning and classification.

Attribute Extent –Subsetting

Subset type events mean that some subset of the data has been singled out for special attention. In the GeoJabber, the names of the events that have been implemented and that fall into this category are selection (often by brushing), indication, subspace, and conditioning. Focus events are supported, via the mechanism of conditioning. The selection event itself is represented by an array of integers, which identifies the selected observations. This array is created by the broadcasting component, and read by all receiving components. The event carries a reference to the array. An example of coordinated selection is shown in Figure 4.

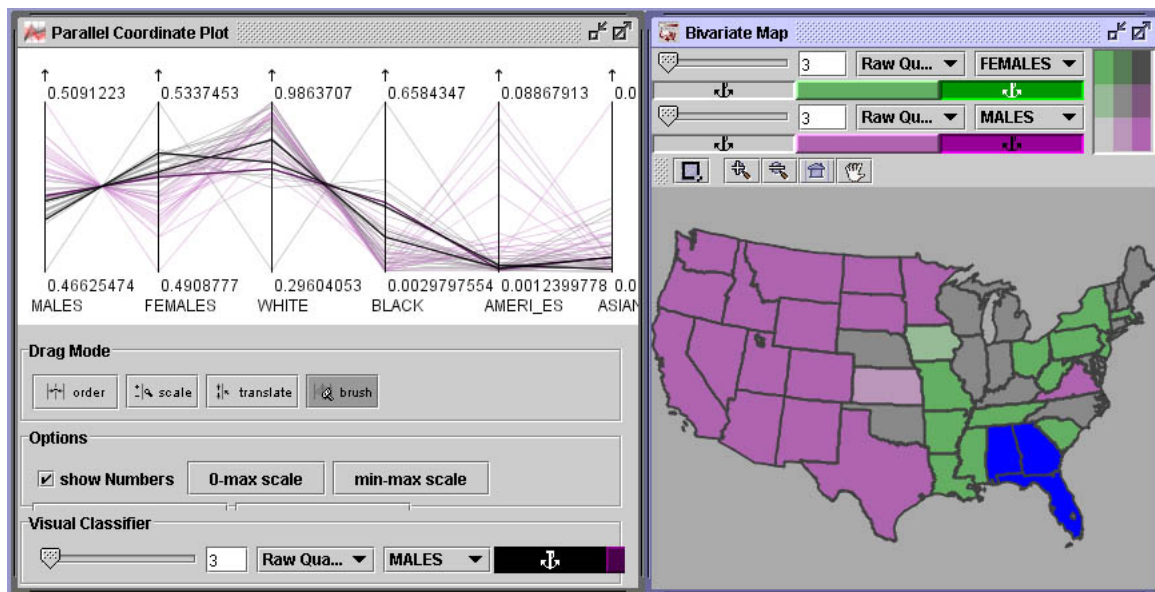


Figure : Coordinated selection of three U.S. States, brushing on the map results in highlighting of the strings for the three selected states in the PCP.

Indication events, in the current implementation, support a single observation. As noted, conceptually, indication can support multiple observations, as in excentric labeling. However, support for representing excentric labeling has been integrated into some of the components of the GeoViz Toolkit, but not into the indication event. Excentric labeling, for now, shows up only in the tool the user is interacting with, but the general model for coordination makes it possible for this event to be coordinated in the future without changing each component independently. Figure 4 shows an instance of indication in action between a map and parallel coordinate plot. The indication in the map is shown by a green hatching on the state of North Dakota. In the PCP, the string associated with North Dakota is highlighted. An example of excentric labeling is also shown (within the map only).

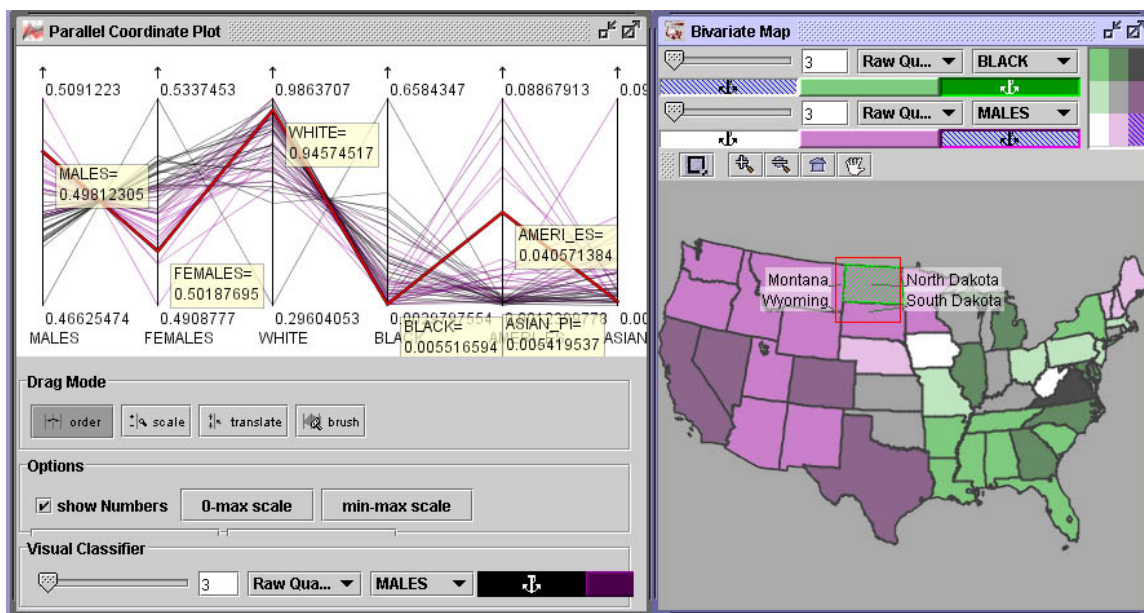


Figure : Coordinated indication, with the indicated state set at North Dakota. Excentric labeling is also shown in the map, but not in the other components, because it is not (yet) supported by indication coordination mechanisms. Note that the indication is also reflected in the visual classifiers and in the bivariate legend.

The subspace events consist of an array of integers, which give the indexes of the currently active columns of data to analyze. The order of the integers is significant; it informs the visualization components of the order in which the data variables should be displayed.

Spatial type events

The only spatial type event currently supported is the spatial extent event. This event type is broadcast by instances of geographic maps. It is useful to keep the same geographic area in view among multiple instances.

The coordinator allows this to be turned on or off, so that it does not force all maps to show the same extent. As a result, it is possible to have both overview and detail (zoomed) maps in the same application. Statistical visualization components, like Parallel Coordinate Plots, can also potentially support these changes in extent. This would require either making the PCP spatially aware, or would require adding information to the event, so that clients could find out which observations are within the spatial area. One difference between this type of event and a selection event is that it requires different symbolization in many clients; for example, a map might draw boundary lines with a wider stroke at a given extent. Other components may want to express different spatial extents with alternative symbolization as well, for example, when a map zoom causes a county-level depiction to change into a state-level depiction, the PCP might do the same.

Attribute extent – focus and conditioning

Conditioning is a filter on the dataset. If the filter is applied to the data variable currently being displayed, this is considered a focus event. If the filter is applied to another variable, this conforms to the usual definition of conditioning. Figure 6 shows a map, a PCP, and a device called the conditioning manager that provides conditioning facilities. Similar to the manner in which selection is implemented, conditioning is represented by a shared array of integers. The events broadcast the array to the listening components; the listening components receive the reference, and then read the array.



Figure : Coordinated conditioning, showing the U.S. states with low proportions (between 4.6% and 7.0%) of divorced people. The map is a bivariate map of %black and %male. States not within the

conditioning filter for %divorced are shown in the background color. The PCP displays strings only for states within the conditioning filter.

Classifying type events

Currently, there are two types of events that fall into the classifying type. One is the ColorClassifier event the other the classification event. ColorClassifier events are described in the previous section, because they fall into both the symbolization and classification categories.

A classifying event contains a reference to a *classifier*. A classifier, defined by a simple interface, is a class that can return an array of integers, when given an array of numeric data and a number of classes. This array of integers shows what class the given numbers fall into. For example, if a classifier is given the numbers [23.4, 0, 2.1, 34.2], and asked to classify it into two classes, it might return the array [1, 0, 0, 1], signifying that the middle two observations fall into the first class, and the first and last observations fall into a second class. This simple contract can be fulfilled by many types of classification methods. Other *Studio* developers have implemented several such classifiers, three of which are supported by GeoJabber.

Future additions

In the future, the following are important event types that should be added. For all of these event types, the difficulty is less in finding an appropriate representation form for the event, and more in providing the necessary support in all associated components. Firstly, more types of subsetting should be allowed, i.e. intersection, xor selection, etc. Fuzzy sets would be a logical and potentially powerful extension. Secondly, events that describe an extension to the current data set should be included, i.e. a new attribute being added for each existing observation. Thirdly, symbolization should be expanded to represent size of symbol, shape, and other visual (and dynamic and sonic and haptic) variables as well as colors. Map algebra type events will be supported in the future. Additionally, temporal extent and temporal synchronization events need to be supported.

GeoJabber

The various operations are realized in the context of collaborative geographic visualization by creating events which encapsulate these operations, and then attaching them to Jabber, also known as XMPP, packets.

Jabber is an open standard for instant messaging (Chatterjee et al. 2005). It is based on XML, in that valid XMPP packets are also well-formed XML packets. XMPP specifies an extension mechanism which is used here to attach the visualization events to the XMPP packets. Java objects representing the various visualization aspects are have been marshaled to XML using the Open Source XStream library (Walnes and Schaible 2008). For simple objects, no mapping is required. For complex objects (including all of the components in the GeoViz Toolkit) this requires writing small adapter classes for each type to be marshaled to XML and back to Java. This process of turning “live” software objects into XML text, and then back again, is a key ingredient in making GeoJabber able to handle arbitrary types of coordination. An advantage of this approach is that the various components do not need to distinguish between a “normal” event, and one that is from the GeoJabber channel. The GeoViz Toolkit with the GeoJabber tool enabled is shown in Figure 7.

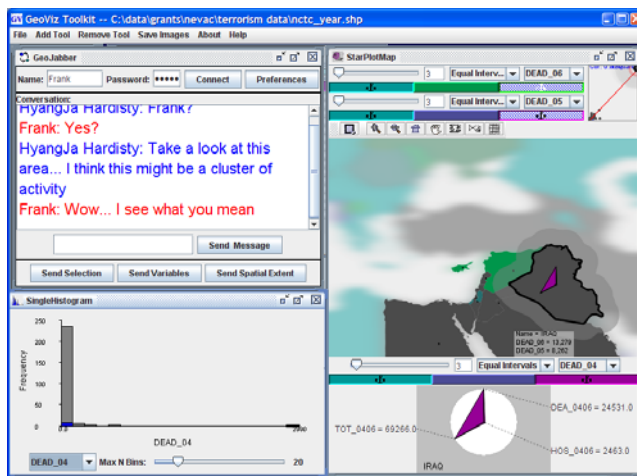


Figure : Coordinated selection of three U.S. States, brushing on the map results in highlighting of the strings for the three selected states in the PCP.

Next, various approaches to collaborative visualization that were evaluated are described, in order of attractiveness, from most suitable to least suitable.

Jabber (XMPP)

Jabber was the technology selected for implementation. It has the virtue of language and platform independence, and its basis in XML means that it is human-readable. There are many XML transformation and storage routines

freely available, which makes it easy to work with. A disadvantage is that XML is verbose: an array which would take four megabytes in binary form might take thirty two megabytes in XML form. Compression can help with this problem, but that makes the XML binary, and if the data structure is large, such as a detailed map of US counties, the data may not fit in main memory.

Jabber packet structure

```
<message from='juliet@capulet.com'  
    to='romeo@montague.net'  
    id='message22'>  
  <body>  
    Wherefore art thou, Romeo?  
  </body>  
</message>
```

Figure : Canonical Jabber message packet

The canonical Jabber message packet example is shown above.

Note that it is fairly easy to read, even without extensive prior knowledge of Jabber or XML. Since it is XML, further information, such as representations of the software objects described earlier in this paper, can be included in the packets. The XML representation of the software objects is generated using XStream, as described above.

RMI

RMI was the runner-up implementation strategy. A working prototype was created using RMI, but was abandoned for the following reasons: It is a Java only solution, it is brittle across changes to clients and servers, and it requires a three-tier system of registries, servers, and clients, where each coordinating party has to be both a client and a server. This is too much technical and conceptual overhead for this part of the application.

Other alternatives evaluated for this purpose were JXTA (a peer-to-peer technology developed by Sun Microsystems) and raw sockets. JXTA is interesting but not mature enough for current use. Raw sockets do not support the concept of objects and are thus too low-level.

Conclusions and Future Work

This paper described the GeoViz Toolkit, a typology of the types of operations supported by the Toolkit, and an implementation strategy for using the XMPP standard for same-time collaboration between distributed users. The types of visual operations described will be coordinated over XMPP.

To better serve analysts, practitioners of visual analytics need to better understand how visual analytic tools are used. The field of human-computer interaction (HCI) has approached this problem (in part) by examining human-computer interaction events, such as mouse motion, or tool activation, which are automatically produced during tool use. One limitation of this method has been that it is difficult to extract the meaningful parts of such event streams. The resulting software can be used as a test bed for experimenting with collaborative functionality in a geospatial context. The peer-to-peer architecture allows the possibility of a secure and extensible system. It is available to users in both source code and executable forms. The source code is available from <http://code.google.com/p/geoviz/source/checkout> via the svn (subversion) protocol, and the latest executable can be run from <http://www.geovista.psu.edu/grants/cdcesda/software/>.

Another issue of interest is that if multiple users make coordinated changes to the map interface, for example the current extent being viewed, it is potentially disorienting to the map user. Imagine two users simultaneously trying to zoom in on different areas of a map. If the users' maps are fully coordinated in the spatial extent that the maps show, the users might experience frustration and conflict. Two alternative options for mediating the conflict are having persistent settings and to have a selectable history of map extents. The persistent setting approach is to have "Leader" and "Follower" settings on each map. If one user sets his or her map to the "Leader" setting, and the other user sets his or her map to the "Follower" setting, the leaders' map extent would automatically be reflected in the followers map. The alternative, selectable history, method would be to give each user a clickable

list of spatial extents, and allow the user to click on them to apply them. These approaches have shown promise in other problem domains (Balkanski and Hurault-Plantet 2000).

References

- Albrecht, J. 1999. Geospatial information standards. A comparative study of approaches in the standardisation of geospatial information. *Computers & Geosciences* 25 (1):9-24.
- Albrecht, J., and H. Kempainen. 1996. A Framework for Defining the New ISO Standard for Spatial Operators. *GIS Research UK*:78-84.
- Balkanski, C., and M. Hurault-Plantet. 2000. Cooperative requests and replies in a collaborative dialogue model. *International Journal of Human-Computer Studies* 53 (6):915-968.
- Becker, R.A., and W.S. Cleveland. 1987. Brushing Scatterplots. *Technometrics* 29:127-142.
- Bertin, J. 1981. *Graphics and Graphic Information Processing*: Springer-Verlag.
- Carr, Daniel B., John F. Wallin, and D. Andrew Carr. 2000. Two new templates for epidemiology applications: linked micromap plots and conditioned choropleth maps. *Statistics in Medicine* 19:2521-2538.
- Chatterjee, S., T. Abhichandani, H. Q. Li, B. Tulu, and J. Byun. 2005. Instant messaging and presence technologies for college compuses. *Ieee Network* 19 (3):4-13.
- Cleveland, W. S. 1987. Research in Statistical Graphics. *Journal of the American Statistical Association* 82 (398):419-423.
- Crampton, J. 2002. Interactivity Types in Geographic Visualization. *Cartography and Geographic Information Science* 29 (2).
- DiBiase, D., A. M. MacEachren, J. B. Krygier, and C. Reeves. 1992. Animation and the role of map design in scientific visualization. *Cartography and Geographic Information Systems* 19 (4):201-??
- Egbert, S. L., and T. A. Slocum. 1992. Exploremap - an exploration system for choropleth maps. *Annals of the Association of American Geographers* 82 (2):275-288.
- Fekete, J., and C. Plaisant. 1999. Excentric Labeling: Dynamic Neighborhood Labeling for Data Visualization. Paper read at CHI'99, May 15-20, at Pittsburgh, PA, USA.
- Fisher Keller, M., J. Friedman, and J. Tukey. 1974. Prim-9: An Interactive Multidimensional Data Display And Analysis System. Stanford, California: Stanford Linear Accelerator Center.
- Griffin, Amy. 2000. Feeling it Out: The Use of Haptic Visualization for Exploratory Geographic Analysis. *Cartographic Perspectives* 39:12 - 29.
- Heiser, W. *Journal of Classification* 2003 [cited. Available from <http://link.springer.de/link/service/journals/00357/about.htm>].
- Hurley, C. B. 2000. Graphical selection of data views. *Journal of Computational and Graphical Statistics* 9 (3):558-581.
- Keim, D. A. 2002. Information Visualization and Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics* 8 (1).
- Keller, P., and M. Keller. 1992. *Visual Cues: Practical Data Visualization*. Los Alamitos, CA: IEEE Computer Society Press.
- Krygier, John. 1994. Sound and geographic visualization. In *Visualization in Modern Cartography*, edited by A. MacEachren and D. R. F. Taylor. New York: Pergamon.
- MacEachren, A. M. 1995. *How Maps Work: Representation, Visualization, and Design*. New York: Guilford Press.
- Mak, K., and M. R. C. Coulson. 1991. Map-User Response to Computer-Generated Choropleth Maps - Comparative Experiments in Classification and Symbolization. *Cartography and Geographic Information Systems* 18 (2):109-124.
- North, C., and B. Shneiderman. 2000. Snap-together visualization: can users construct and operate coordinated visualizations? *International Journal of Human-Computer Studies* 53 (5):715-739.
- NRC. 2003. *IT Roadmap to a Geospatial Future*: National Academies Press.
- Pietzuch, Peter R., and Jean Bacon. 2002. Hermes: A Distributed Event-Based Middleware Architecture. Paper read at The 22nd International Conference on Distributed Computing Systems.
- Qian, L., M. Wachowicz, D. Peuquet, and A. MacEachren. 1997. Delineating Operations for Visualization and Analysis for Space-time Data in Gis. Paper read at GIS/LIS '97, at Cincinnati, OH.
- Raskin, Jef. 2000. *The Humane Interface*. Reading, MA: ACM Press.
- Walnes, Joe, and Jorg Schaible. *XStream* (1.3.1) 2008 [cited. Available from <http://xstream.codehaus.org/>].
- Ware, Colin. 2000. *Information Visualization: Perception for Design*. Edited by S. K. Card, J. Grudin, J. Nielsen and T. Skelly, *The Morgan Kaufmann Series in Interactive Technologies*. San Diego, CA: Academic Press.
- Whigham, P. A., R. I. McKay, and J. R. Davis. 1992. Machine Induction of Geospatial Knowledge. *Lecture Notes in Computer Science* 639:402-417.
- Wilkinson, L. 1999. *The Grammar of Graphics*. New York, NY: Springer-Verlag.