Geographical information systems technologies for spatial visualization of statistical data

Dardala, Marian and Reveiu, Adriana

Academy of Economic Studies Bucharest, Academy of Economic Studies Bucharest

2 May 2011

GEOGRAPHICAL INFORMATION SYSTEMS TECHNOLOGIES FOR SPATIAL VISUALIZATION OF STATISTICAL DATA

Marian DÂRDALĂ
Academy of Economic Studies, Bucharest
dardala@ase.ro
Adriana REVEIU
Academy of Economic Studies, Bucharest
reveiua@ase.ro

Abstract. *This paper proposes an original solution to present statistical data using the facilities provided by the Geographical Information Systems, to improve the means of statistical data figure and distribution inside the territorial profile. The proposed solution allows to represent more statistical data sets, on the same map, using two different methods of data representation: using a color ramp to represent the territorial distribution of an aggregate indicator and a set of charts overlapped to represent the proportions of the variables that form the indicator.*

1. Introduction

In recent years, Geographical Information Systems (GIS) have become an important tool for statistical data representation.

The huge amount of spatial data generated by GIS expansion, the increasing number of GIS applications available, the computerization of a large amount of information sources, and the availability of digital map has increased the opportunity and need for the usage of methods for spatial visualization of statistical data, for both research and applied purposes. Data graphical representation in GIS requests to use different symbologies to view more information in the same time on the map. The spatial perspective of data offer the advantage to locate the studied phenomena and the symbology offers quantitative details that are visual reflected in various forms.

2. Techniques for Statistical Data Visualization in Geographical Information Systems

Cartograms are unique representations of geographical space. A cartogram is a type of graphic tool that describes attributes of geographic objects. Mapping requirements include the preservation of shape, orientation contiguity, and data that have suitable variation. Successful communication depends on how well the map reader recognizes the shapes of the internal enumeration units, the accuracy of estimating these areas, and effective legend design. Because a cartogram does not depict geographic space, but rather changes the size of objects depending on a certain attribute, a cartogram is not a true map. Cartograms vary on their degree in which geographic space is changed: some appear very similar to a map; however some don't look like a map. (Borden, 1996)

There are three main types of cartograms, each having a very different way of showing attributes of geographic objects: non-contiguous, contiguous and Dorling cartograms. A **non-contiguous cartogram** is the simplest and easiest type of cartogram. In a non-contiguous cartogram, the geographic objects do not have to maintain connectivity with their

adjacent objects. This connectivity is called topology. By freeing the objects from their vicinity, the objects can grow or get smaller in size and still maintain their shape.

In a **contiguous cartogram** is maintained the true topology, the objects remain connected with each other, but this causes great distortion in shape. In this case, the cartographer must make the objects of the appropriate size, to represent the attribute value, but he or she must also maintain the shape of objects as best as possible, so that the cartogram can be easily interpreted.

A **Dorling cartogram** maintains neither shape topology nor object centroids, though it has proven to be a very effective cartogram method. This type of cartogram was named after its inventor, Danny Dorling of the University of Leeds. To create a Dorling cartogram, instead of expanding or distorting the objects themselves, the cartographer will replace the objects with a uniform shape, usually a circle of the appropriate size.

3. Informatics Technologies for Representing Symbologies in Geographical Information Systems

A geographic information system works with spatial data. These data are used to represent different elements of geographical reality, the types of vectors (spatial characteristics) and their location on the earth. The main vectorial spatial data types are: point, line and polygon.

In addition to the vectorial spatial data, geographical information systems operate with raster data. They are described by a grid of color (bit maps), covering a well defined area on the land area and are used to suggest more realism to the studied geographical space. Satellite images, terrain elevation model (DEM), orthophotoplan, and scanned maps are represented using raster formats.

The geo-relational data model involves the data storage using independent formats based on the division of attributes into two entities, and store them separately. Thus, a spatial data is made up of a spatial attribute, which is stored separately and non-spatial attributes. The link is made through an identification field (ID), which aims to combine consistently the two types of attributes. Spatial data storage using geobase data model supposes to store both the spatial attributes and the non-spatial attribute, together in one table. The field containing the space attribute is a geometric, a vector field.

The spatial features are implemented and processed through all classes of objects even they reside in tables, in a relational database. Thus, they contain properties and methods that operate on them.

3.1. ArcObjects .NET SDK Framework for Geographical Information Systems Development

ArcObjects represents a framework based on COM (*Component Object Model*) used to build ArcGIS software product, developed by ESRI (*Environmental Systems Research Institute*). The three versions of ArcGIS software, ArcView, ArcEditor, and ArcInfo share the same applications: ArcCatalog and ArcMap. Geobase data model and ArcObjects are the foundation of the two desktop applications.

ArcObjects is composed from classes and objects. ArcObjects uses three types of classes (Chang, 2008):

- **Coclasses** – used to create new objects, like *RgbColor* class having as the goal to define and to handle colors, defined in RGB model.
- **Abstract classes** -cannot be used to generate new objects, but are designed to build hierarchy of classes starting with an entity having a general nature. For example, classes that implement different color models are derived from an abstract class called *Color*;

- **Classes** – are not used directly to generate new objects, but they are build from other objects. For example, to access selected features on the map at a time, a *MapSelection* object can be build from the map object.

Using ArcObjects, we can develop applications in several ways:
- GIS applications can be developed using the independent component *ArcEngine* and a software development environment like *Visual Studio.NET*;
- It is possible to develop extensions, which could be attached to the ArcGIS applications and which contains interaction elements for different customized GIS processing;
- It is possible to develop Web applications.

3.2 Symbologies for Statistical Data Representation

One of the main applications of GIS is related to the possibility to display various statistical information on the maps. So that, the users could have a geographical perspective on the represented phenomenon, too. We must build cartograms to display the results of the economic and social phenomena.

Another kind of symbology refers to the displaying of graphs on the spatial data, to capture the proportions of variables represented as statistical data.

There are several techniques implemented in ArcGIS, used to classify some items according with the value of one indicator (attribute). To achieve this, it's a must to identify disjoint intervals of values and the indicator used to form groups. For example, the simplest way to create ranges of values is the division of the values used to make the classification, into equal intervals. The *NaturalBreaks* method, used to split the ranges, is based on the shaping of the groups of values, so that to have a minimum dispersion inside the group and a maximum dispersion between groups.

In ESRI's software, colors could be defined individually. In this case there are special classes for various color models, like *RgbColor* class used for RGB model, and as collections of colors, managed by color ramp classes, having the name *ColorRamp*. Collections of colors can be generated by the programmer or can be used the default collections that are defined by ESRI's style gallery.

In the following paragraph we present the exemplifications of the usage of these symbologies.

4. Results

From the relational model perspective, we create a personal geodatabase *bd_ro.mdb*, with *judete* table containing Romania's counties as spatial data and *populatie* table containing the data about the population of the counties, as follows:
- population in large towns (with 150000 inhabitants or more);
- population in towns (having between 10000 and 150000 inhabitants);
- rural population (with less than 10000 inhabitants).

The two tables contain a common field used to create the junction between them; it is the symbol of the county.

The first part of the statistical data representation supposes to color the counties on the map in a gradient color, based on their total population. Thus, by pressing the button from the toolbar built for this purpose, the map is colored based on the total population and the appropriate legend will be displayed too.

The source code contains several parts according to the steps that solves. In this paper we present only the specific steps:
- To identify the desired thematic layer - the counties layer:

```
int nl, i, nrg=3; // nl-variable used to store the number of layers
// i –counter variable, nrg – number of groups (ranges) used to divide the counties, function on
```

```
// the population number
IMxDocument doc = ArcMap.Document; // get the map document
IMap harta = doc.FocusMap; // get the map
// declare the interface for the layer – each contains properties and methods needed in different
// contexts
ILayer lyr = null;
IFeatureLayer ifl = null;
IFeatureClass ifc = null;
// getting the number of layers from the map
nl = harta.LayerCount;
// iteration of the layers to identify the desired county
for (i = 0; i < nl; i++)
{lyr = harta.get_Layer(i);
   if (lyr.Name == "judete")
   { // the usage of IFeatureLayer  interface for the desired layer
      ifl = (IFeatureLayer)lyr;
// using IFeatureLayer interface we can access FeatureClass property which contains properties
// and methods of the class with spatial characteristics
      ifc = (IFeatureClass)ifl.FeatureClass; }}
```

- Building the intervals for counties classifying according with the population number.

```
// building a histogram based on the attribute used to make the classification
ITableHistogram ith = new BasicTableHistogramClass();
ith.Field = "Total_populatie"; // the field
ith.Table = tpop; // table with the field
object vval, vfrecv;  // vval - attribute value, vfrecv - frequencies
 ((IBasicHistogram)ith).GetHistogram(out vval, out vfrecv); // loading objects
// building an interface for classification, based on NaturalBreaks method
IClassify div_grupe = new NaturalBreaksClass();
// assigning data used to generate the ranges
div_grupe.SetHistogramData(vval, vfrecv);
div_grupe.Classify(ref nrg);  //loading the number of ranges which will be generated
double[] vecgr; //declaring an array with the ranges limits
vecgr = (double [])div_grupe.ClassBreaks; // limits generation
```

- Coloring the map using colors from the color ramp and the classification in the three ranges of values, previously generated.

```
// defining an object for a symbology based on the ranges of values
IClassBreaksRenderer cbr = new ClassBreaksRenderer();
cbr.Field = "Total_populatie";// the field which contains the values used to color the map
cbr.BreakCount = nrg; // the number of range
// setting the minimum value, which is the lowest limit of the first range of values
cbr.MinimumBreak = vecgr[0];
// declare an interface which shows the way to fill the polygons with color
ISimpleFillSymbol fs;
for (i = 0; i < cbr.BreakCount; i++)
  {
    fs = new SimpleFillSymbol();
    fs.Color = rcul.get_Color(i); // getting filling color
    cbr.set_Symbol(i, (ISymbol)fs);    // associate the filling symbol
    cbr.set_Break(i, vecgr[i+1]);     // setting the highest limit of the range
```

```
    cbr.set_Label(i, vecgr[i].ToString()+ "-" + vecgr[i+1].ToString() ); // labeling in the legend of the
//map from Table of Content
  }
```

IGeoFeatureLayer gfl = (IGeoFeatureLayer)lyr;

// associate the symbology (*cbr*) to the *lyr* layer, corresponding to the counties, by *Renderer* property

gfl.Renderer = (IFeatureRenderer)cbr;

doc.ActiveView.Refresh();// redraw the content of current view

doc.UpdateContents(); // update the content of the Table of Content window

This symbology is used as background for a chart symbology that is presented in the next item.

- The next step is to display the charts on the spatial data to figure the proportions of the variables inside the statistical data represented. To exemplify this, we choose to represent the proportion of the population from the cities, towns and rural areas in total population, for all counties, on the map using pie type charts, as shown in Figure 1. Data about the population, in counties, cities, towns and rural population are stored in the *populatie* table, from the *bd_ro.mdb* database. To avoid the code duplication code comparing with the previous example, we will present only the specific code sequences.
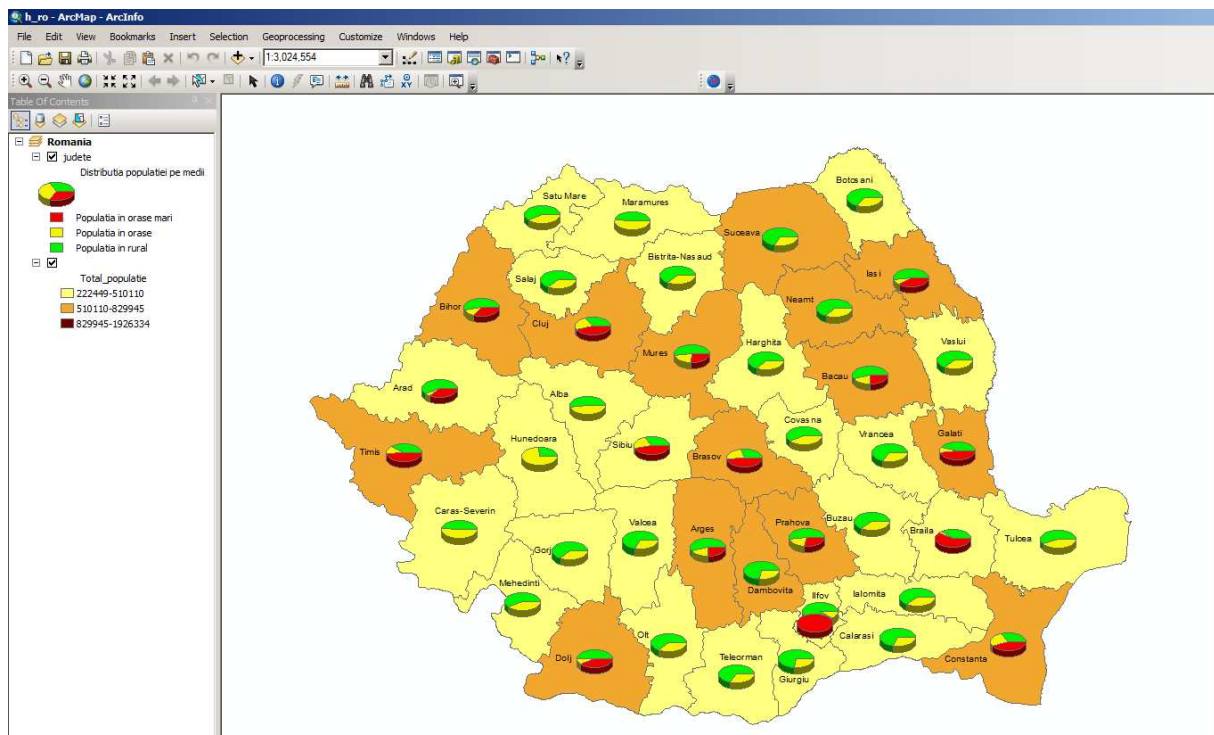

Figure 1 Colouring the map using the color ramp and the PieChart symbology based on counties' population

- To effectively display symbology of *piechart* type, on the thematic layer *counties* from the map, it is necessary to do as following:

// building a color ramp of nrg (= 3) colors, using Red to Green color ramp from ESRI galery,

IColorRamp rcul = null;

EnumStyleGalleryItem vst = doc.StyleGallery.get_Items("Color Ramps", "ESRI.style",

"Default Ramps"); //doc= map document

 IStyleGalleryItem elem = vst.Next();

while (elem != null)

  { if (elem.Name == "Red to Green")

    { rcul = (IColorRamp)elem.Item;

```
      break;}
    elem = vst.Next();}
rcul.Size = nrg;
bool vb;
rcul.CreateRamp(out vb);
ISimpleFillSymbol fs; //defining an interface to describe the filling types
IPieChartSymbol pcs = new PieChartSymbolClass(); //building an interface for piechart symbology
// building the interface to display the chart type symbology
IChartRenderer cr = new ChartRenderer();
pcs.Clockwise = true; // setting the direction for building of the pie chart
 ((IMarkerSymbol)pcs).Size = 30; // the symbol size (for the piechart)
// the parameters for the nrg (=3) variables which compose the piechart
for (i = 0; i < nrg; i++)
  { // building a symbol for chart filling
    fs = new SimpleFillSymbol();
    fs.Color = rcul.get_Color(i);    // associate the filling color taken from the color ramp previous build
    ((ISymbolArray)pcs).AddSymbol((ISymbol)fs);    // adding the symbol to the collection of symbols
    ((IRendererFields)cr).AddField(numec[i]);    // adding the field used to create the piechart
    // associate the field label used to display the legend
    ((IRendererFields)cr).set_FieldAlias(i, vs[i]); }
cr.ChartSymbol = (IChartSymbol)pcs; // associate the piechart type symbology
cr.CreateLegend(); // building the legend – will be dispayed in the Table of Content
cr.Label = "Distributia populatiei pe medii"; // labeling the symbology in the legend
cr.UseOverposter = true; // avoidance the charts overlapped
// associate the symbology (cr) to the lyr layer, corresponding to the counties
IGeoFeatureLayer gfl = (IGeoFeatureLayer)lyr;
gfl.Renderer = (IFeatureRenderer)cr;
doc.ActiveView.Refresh(); // redraw the content of the current view
doc.UpdateContents(); // updated the content of the Table of Content window
```

5. Conclusions

The proposed solution involves two symbologies that are overlaid and that reflect in two ways the same thing. The Class Breaks symbology reflect the total population indicator and plays role of background for a Chart symbology that reflect de proportion of the different categories of population in total population for each county. This type of cartogram is suitable in case of large enough regions because in this case the cartogram is more visible and the users easily can distinguish the results displayed by cartogram.

Acknowledgments

References

1. ***, "Exploring ArcObjects", ESRI Press, 2002;

2. Burke, R., "Getting to Know ArcObjects: Programming ArcGIS with VBA", ESRI Press, 2003;

3. Borden D. D., "Cartography: Thematic Map Design", 4th edition, William C Brown Publishing House, 1996.

4. Chang, K.-T., "Programming ArcObjects with VBA", CRC Press, 2008;

5. Ioniţă, C., "Collaborative Business Process Optimization Using Domain Specific Languages", Proceedings of the IXth International Conference on Informatics in Economy, Bucharest, 2009.

Adriana REVEIU has graduated the Faculty of Cybernetics, Statistics and Economic Informatics. She holds a PhD diploma in Economic Cybernetics and Statistics. She is associate professor in Economic Informatics field and branches within Department of Economic Informatics at faculty of Cybernetics, Statistics and Economic Informatics from Academy of Economic Studies. She is the author and co-author of 10 books and over 50 articles in journal and proceedings of national and international conferences, symposiums, workshops in the fields of multimedia, GIS, communications, learning systems and data management.

Marian DÂRDALĂ has graduated the Faculty of Cybernetics, Statistics and Economic Informatics. He holds a PhD diploma in Economic Cybernetics and Statistics. He is professor in Economic Informatics field and branches within Department of Economic Informatics at faculty of Cybernetics, Statistics and Economic Informatics from Academy of Economic Studies. He is the author and co-author of 11 books and over 50 articles in journal and proceedings of national and international conferences, symposiums, workshops in the fields of multimedia, human computer interaction - HCI, GIS, data bases and object oriented programming.