355

INT. J. GEOGRAPHICAL INFORMATION SCIENCE, 2002
VOL. 16, NO. 1, 41–53

Taylor & Francis
Taylor & Francis Group

Research Article

Integrating spatial data analysis and GIS: a new implementation using the Component Object Model (COM)

MATTHEW J. UNGERER and MICHAEL F. GOODCHILD
National Center for Geographic Information and Analysis, Department of Geography, University of California, Santa Barbara, CA 93106-4060, USA;
e-mail: unj@geog.ucsb.edu, good@geog.ucsb.edu

(*Received 18 December 2000; accepted 19 July 2001*)

Abstract. This paper presents a coupling strategy based on Component Object Model (COM) technology, for performing spatial analysis within a GIS. The strategy involves using a module which simultaneously manipulates software components from the GIS application and the data analysis application. We illustrate the strategy using an extension, written for the proprietary GIS ArcInfo, which performs areal interpolation, a statistical method of basis change commonly required by users of socioeconomic data. The extension creates an instance of a statistics package and uses it to process GIS data stored in ArcInfo, and then passes the resulting information back to ArcInfo where it is stored in a standard attribute table. This coupling strategy can, of course, be used with other COM-compliant GIS and data analysis software. COM-compliant software allows GIS analysts and researchers to create custom-tailored applications using components from many different sources. Because the GIS does not rely on a proprietary macro language for customization there is a potential increase in access to spatial analysis tools which were previously difficult to link with a GIS, and we explore and evaluate the potential of the coupling strategy presented here for the GIS and spatial analysis research community.

1. Introduction

One of the most popular research areas in geographical information science is the development of new spatial analytical techniques and the incorporation of both new and existing techniques within a GIS environment (Fotheringham and Rogerson 1998, Getis 2000, Marble 2000). In this paper we illustrate a coupling strategy for performing spatial analysis within a GIS, based on Component Object Model (COM) components, and position it along the continuum of coupling strategies. We then illustrate the new coupling strategy using an areal interpolation example. The specific strategy for performing areal interpolation is not a new one, but the coupling strategy is a unique method that we hope will prove useful in the integration of spatial statistics and GIS.

This coupling strategy is based on COM components, and is implemented using one module of code which uses the software components of both the analysis software and the GIS (see figure 1). The code module programmatically opens the analysis

International Journal of Geographical Information Science
ISSN 1365-8816 print/ISSN 1362-3087 online © 2002 Taylor & Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/13658810110095066

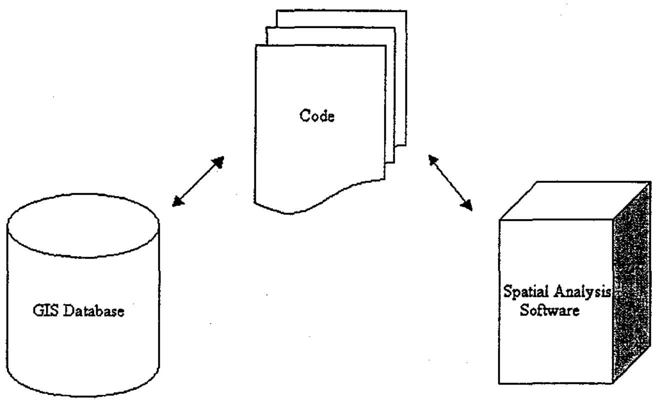

Figure 1. A conceptual diagram of the coupling strategy.

software package, transfers the data from the GIS, performs the analysis, transfers the data back to the GIS and finally closes the analysis software package. The example in this paper exploits the fact that the proprietary GIS ArcInfo v.8.0.2 and the Microsoft Excel 2000 spreadsheet package can both be customized by any programming language compatible with the COM protocol. Other GIS, statistical analysis and related software packages which support some degree of COM-compliant customization include the proprietary GIS Idrisi (http://www.clarklabs.org), the S-PLUS statistics package (http://www.splus.mathsoft.com) and the SWARM object-oriented modelling environment (http://www.swarm.org).

Other researchers have explored linkages using the COM protocol. Zhang and Griffith (2000) implemented a linkage between a Microsoft Access database and ESRI's MapObjects software which enabled them to use GIS components from within the database software. The linkage which we implemented differs in that we are using statistical components from within the GIS. Our approach is similar to the one taken by Bao *et al.* (2000). In their design, which takes place within the GIS, components are used to pass spatial attributes from the GIS to analytical software which in turn passes the results back to the GIS. The main distinction is purely technical: in our approach the same software module contains objects from both the GIS and the analysis software. Both approaches enable components from different software packages to access the same data, and thus give the analyst access to many standard statistical routines which have already been written and refined.

2. Background

This section summarizes the current coupling strategies between spatial statistics and GIS and describes the COM protocol.

2.1. *Integration of spatial statistical methods and GIS*

Following Bailey (1998), we define 'spatial analysis' as *statistical* spatial analysis, thus drawing a distinction between classical statistical techniques modified as necessary for spatial data and other 'analysis' methods commonly found in commercial GIS packages (e.g. site allocation, network analysis and cartographic modelling). In many ways a GIS is the ideal environment in which to perform both exploratory and confirmatory spatial analysis because a GIS is designed to handle data with a geographical footprint. That is to say, data in a GIS are expected to be geographically referenced (their locations are stored either in latitude/longitude coordinates or in some projected coordinate system), a property which is obviously unique to spatial data.

Contemporary commercial GISs already contain sophisticated data manipulation (which is useful for data pre-processing before an analysis), input, output and display functionality. In addition, the databases of many GISs explicitly contain the topological relationships between various spatial features. Finally, a wealth of spatial data already exists in GIS databases, and it is convenient to be able perform the analysis from within the software in which the data are typically accessed. At least three major sources of GIS data on the world wide web now contain over a terabyte of information: the Alexandria Digital Library (http://webclient.alexandria.ucsb.edu), the EROS Data Center (http://edcwww.cr.usgs.gov), and Terraserver (http://terraserver.microsoft.com).

To be used in a commercial spatial data analysis package such as the S-PLUS SpatialStats module, data must first be converted to an appropriate input format. Commercial spatial data analysis packages are typically not as sophisticated at data input, output and display as commercial GISs, so data must be reconverted to the GIS's format for use. This creates extra unnecessary work for the GIS analyst, especially if an analysis must be repeated on multiple datasets. Thus the arguments for the provision of spatial statistical analysis functions within a GIS are many.

Much progress has been made on integrating spatial data analysis and GIS, and a variety of techniques are currently available at some level of integration, including simple K-function computation (Rowlingson *et al.* 1991), measures of spatial autocorrelation (Kehris 1990a, Ding and Fotheringham 1991), simple regression modelling (Kehris 1990b), and geostatistical interpolation via kriging (Pebesma and Wesseling 1998, ESRI 1999). The SpaceStat software developed by Anselin (2000) contains a variety of spatial analysis routines and is available as both an ArcView extension and as separate freestanding software. Most of these spatial analytical techniques are not available directly as commands from within the GIS, with the exception of the geostatistical and a few other analysis routines. This is because proprietary GIS development parallels market demand and the current demand for spatial analytical functions, although growing, is still relatively low. Consequently, most proprietary systems have focused on GIS development in other areas. This leads to the need for flexible and adaptive coupling strategies between spatial analysis software and the GIS.

The types of coupling strategies or linkages between spatial analysis software and GIS are summarized in table 1. These are very similar to the linkages between spatial modelling software and GIS (Batty and Xie 1994a, 1994b, Park and Wagner 1997). The available linkages are to a certain degree dependent upon the linkages and customization options provided by the GIS vendor. For instance, if the GIS does not support the ability to export an attribute table in tab-delimited text form, then loose coupling may not be an option.

The analyst implementing a new method of spatial analysis within a GIS must choose which coupling strategy is appropriate by weighing performance, development time, intended users, and other considerations. In many cases close coupling is the best available option for linking spatial analysis and GIS. This is because the analysis

Table 1. Coupling strategies for linking spatial analysis and GIS (adapted from Goodchild *et al.* 1992). The new coupling strategy based on COM components, described in this paper, is a subtype of close coupling.

Strategy	Isolated	Loose	Close	Integrated
Description	Analysis and output display directly in spatial analysis software	Analysis in spatial analysis software, output display in GIS, facilitated by online file exchange	Analysis method varies; GIS and analysis package share a common database	Analysis and output display directly within GIS
Advantages		Little overhead in terms of code creation	Spatial analysis can be done from within the GIS environment	No file import or export, no code creation required
Disadvantages	Abundant GIS data layers cannot be used	Time consuming to import and export data	Overhead in terms of code creation	Possible lack of specialist insight in spatial analysis

can be performed without leaving the GIS environment and because it allows analysts to code the procedure exactly as they wish (Bailey and Gatrell 1995). However, a potential disadvantage of this form of close coupling is that the analyst must write routines that have already been coded many times previously (e.g. a standard linear regression). This type of close coupling is also a disadvantage for the mass of GIS users with no experience in coding spatial statistical methods. This paper exposes a new subtype of close coupling that lacks these disadvantages.

2.2. *Component Object Model protocol*

COM is a standard which enhances software interoperability by allowing different software components, possibly written in different programming languages, to communicate directly (Microsoft 2000). It is important to realize that the COM standard can be implemented for any operating system and for any programming language (the examples in this paper were developed in Visual Basic v.6.0 on Windows NT™). It is a binary standard, which means that different pieces of compiled software that support COM can connect and communicate with each other directly, and also allow the system to be extended without access to proprietary source code.

A user wanting to manipulate components from different sources in the same project simply references the appropriate object libraries to gain access to their components, and then uses the components to perform the desired function. Components are manipulated indirectly through one or more interfaces, which are specifications or contracts to perform a related set of functions. Interfaces are pointers to virtual tables in memory, or 'vtables', which themselves are pointers to different parts of the actual object code. More precisely, an interface is a pointer to a pointer to a vtable, which allows multiple instances of an object to access the same vtable if necessary.

The software components which are exposed in COM implementations are typically shown in Unified Modelling Language (UML) class diagrams, or similar

diagrams, which describe the classes (including relationships between classes), interfaces (including properties and methods), type enumerations, etc. of a particular software package. The ArcInfo and Excel class diagrams show over 850 reusable components between the two packages. Examples of reusable software components in ArcInfo include feature attribute tables, raster band collections (e.g. multi-band remotely sensed imagery), and individual features themselves.

2.3. *VBA as a macro language*

The newest version of ArcInfo is a suite of desktop applications for WindowsNT/2000 with a more graphical user-interface than previous releases. The desktop applications of ArcInfo v.8.x (e.g. ArcMap, ArcCatalog, ArcToolbox) are intended to be customized using any COM-compliant programming language, and Visual Basic for Applications (VBA) is provided with the applications for this purpose. In many ways the VBA development environment is a mature programming language, unlike proprietary macro languages. VBA has become the macro language for multiple software packages because of the degree of user support and the maturity of its development environment, which includes useful features like code completion and an interactive debugger with the ability to display variable values in the current scope. VBA is in many ways a modern object oriented programming language; it has support for over 10 data types (including user-defined data types), automatic type checking, six different repetition structures, multidimensional array constructs, and multithreading. ArcView's Avenue scripting language, a proprietary macro language, lacks most of these features. A robust macro language obviates one of the main reasons for choosing a more standard close coupling method: many analysts and researchers prefer to write their own functions in C or C++ because they are more powerful programming languages than the typical proprietary GIS macro language. If the macro language itself is a powerful development environment then spatial analysis can be performed using the macro language and thus be more fully integrated with the GIS. C and C++ are obviously still more robust than VBA but the gap between them is closing and simpler routines may justifiably be written in VBA. When new spatial analysis routines are written and distributed as macros, rather than compiled source code, the details of the routine's implementation are available to any analyst or researcher using the script. For some purposes, including research, this situation is preferable to the 'black box' approach of standard functions supplied within the GIS.

2.4. *Coupling strategies and prior versions of ArcInfo and ArcView*

Coupling strategies used with prior versions of ArcInfo usually involve the use of Arc Macro Language (AML). While AML is highly functional and robust, it is specific to the workstation Arc/Info architecture. Current trends in the software industry are leading away from proprietary macro languages like ArcView Avenue and AML towards industry standard object-oriented macro languages like VBA. Over 200 software packages now use VBA as their macro language (Microsoft 2000). The emergence of industry standard macro languages has tangible benefits for the average GIS analyst (and for software users in general) because the general programming syntax is the same regardless of the software package, so the user must learn only the new software objects when learning to control a new application. It is important to note that the COM components provided with ArcInfo v.8.0.2 do not support some of the functionality previously available with AML. For instance, when working with the new geodatabase data model, overlays must be performed programmatically by individually overlaying each feature in the input feature class with every other feature in the overlay feature class (sample code is provided with ArcInfo to do this), rather than simply specifying an input feature class and an overlay feature class. It is possible to perform an overlay of two feature classes directly using AML, which is by far the easier operation. However, it is expected that the ArcInfo COM components will eventually provide all of the important functionality presently available in AML.

A coupling strategy similar to the one presented here could be implemented using an older technology, Microsoft's Dynamic Data Exchange (DDE) using, for example, the proprietary GIS ArcView's Avenue scripting language to connect to Excel. DDE allows two applications to engage in what is termed a 'conversation' for data exchange. This particular implementation suffers from two drawbacks. First, the analyst must have knowledge of two macro languages, ArcView's and Excel's, in order to pass an execute request in a conversation. Second, the Avenue scripting language is not nearly as powerful as VBA. For these reasons the new close coupling strategy using COM components is preferred.

Finally, another often-implemented strategy for close-coupling involves the use of dynamic link libraries (DLLs) which are then called from within AML or Avenue scripts. There are perhaps several reasons for calling a routine from a DLL rather than implementing in either AML or Avenue, but two important ones are to add the ability to create more advanced data structures and to improve the speed of complex calculations. For an example of this type of implementation see Anselin and Bao (1997).

3. Methods

To use the COM components of Excel from within an ArcInfo macro, a dynamic link library or a separate executable, it is recommended that an instance of Excel be created using the process of 'early binding' (comments are in *italics*):

```
'Visual Basic code to open MS Excel
Dim appExcel As Excel.Application
Set appExcel=New Excel.Application
appExcel.Visible=False 'Excel is not shown on the screen
```

Early binding simply means that variables are declared with an explicit type and are thus 'bound' to that type. It requires a reference to a DLL or an object library (e.g. a file with a .dll or .olb extension) containing information about the application's objects, properties and methods which can then be used to create a new instance of the application. Some advantages of early binding are a potential improvement in run-time performance, the ability to use the automatic code completion feature and improved code readability. Once the instance of Excel is created, data can be passed back and forth between ArcInfo and Excel to perform an analysis. The data exchange is performed in memory, so that no temporary files need to be written. The last line in the above code makes Excel invisible, which hides from the user the fact that Excel is being used and dramatically improves performance.

To illustrate the coupling strategy, we implemented three areal interpolation routines which are accessed via a command button in ArcInfo. The routines are run from within ArcInfo, but two of them rely on Excel to perform a statistical analysis, either a linear regression or a matrix inversion and matrix multiplication. Most of the code that performs these tasks is actually contained in a Visual Basic form, a graphical document composed of 'controls' (e.g. command buttons, textboxes, etc.). Code associated with the controls is programmed to run in response to user-initiated events. An installer for the extension and source code are available online at http://www.geog.ucsb.edu/~unj/extension.html.

4. An example: areal interpolation

Areal interpolation is a specific form of basis change in which data are reported for one set of areas or zones but are needed for another, independent set. It is a common problem for users of socioeconomic data, which is often reported for political regions that may not have any true spatial significance. Figure 2 shows a hypothetical example of this situation where population data have been collected by county, but is needed by bioregion.

Following Goodchild and Lam (1980), we will use the term 'intensive' to refer to variables that can be expected to have the same value at any location within a homogenous zone as in the whole (e.g. a population density), and 'extensive' to refer to variables that can be expected to have half the zone's value in each half of a homogenous zone (e.g. a population count). A spatially extensive variable can be made intensive simply by dividing by the area of the zone. For this discussion we assume that the variable to be interpolated is spatially extensive (i.e. a count rather than a ratio).

There are several strategies for performing this type of interpolation. Goodchild *et al.* (1993) present a more detailed treatment of the three areal interpolation methods discussed here, while Flowerdew and Green (1998) present alternative strategies. Sadahiro (1999, 2000) reviews and compares several areal interpolation

Figure 2. Source/target overlay areas.

methodologies. One of the simplest strategies, known as the 'areal weighting' or 'piecewise' method is of the form:

$$V_t = \Sigma_s U_s (a_{st}/\Sigma_t a_{st}) \quad (1)$$

where V_t are the target zone values, U_s are the source zone values, and a_{st} are the areas of overlap between the source and the target zones. The density of the variable to be interpolated is assumed to be constant within a source zone, which can be a drawback of this particular areal interpolation method.

The next two methods are generalizations of the piecewise method which allow for auxiliary information to aid in the interpolation process. The second method is of the form:

$$U_s = \Sigma_t d_t \, a_{st} \quad (2)$$

where d_t represents the target zone densities (target zone value/target zone area). The density values can be estimated by inverting the area weight matrix if the number of target zones equals the number of source zones ($n_t = n_s$) or as the coefficients in a linear regression with the constant term set equal to zero if the number of target zones is less than the number of source zones ($n_t < n_s$):

$$y = X\beta \quad (3)$$

Multiplying each target density by the target's area gives the target zone values. This method operates under the limiting assumption of homogeneity in the variable within a target zone, rather than within a source zone.

The third method uses non-congruent control zones within which the intensive form of the variable to be interpolated is assumed to be constant. The intensive form of the variable to be interpolated does not, however, need to follow homogeneity assumptions in either the source or the target zones. The number of control zones is assumed to be less than or equal to the number of source zones ($n_c \leqslant n_s$). In this case the analyst has three layers of data: the source zones containing the value to be interpolated, control zones with unknown but assumed constant value, and target zones for which the value will be estimated. The control zone densities, d_c, can be obtained from:

$$U_s = \Sigma_c d_c \, a_{cs} \quad (4)$$

As before, the d_c values can be estimated by inverting the area weight matrix if $n_c = n_s$ or by a linear regression as previously described if $n_c < n_s$. The target values, V_t, can then be obtained by multiplying the control zone densities by the areas of overlap between the control and the target zone, b_{ct}, and integrating these values over the target zone:

$$V_t = \Sigma_c d_c^* \, b_{ct} \quad (5)$$

All of the data necessary for performing the areal interpolation are available within a standard GIS database. For example, the essential variables for performing the interpolation using the third method presented above are: U_s, a_{cs}, and b_{ct}. In a GIS environment, U_s values are contained in a field in an attribute table and the areas of overlap, a_{cs} and b_{ct}, are easily computed as the result of two GIS overlay operations: the control and source zones and the control and target zones, respectively. The GIS can also determine the number of source, control and target zones,

which it is necessary to know in order to determine which method of solving for the d_c values is appropriate.

Figure 3 shows the Visual Basic form for the areal interpolation extension. The form allows the user to change the areal interpolation method, set the input datasets, specify a size for sliver polygons which are ignored in the analysis, and specify whether to save the spreadsheet that is created by Excel for review or simply discard it.

User-initiated events, such as clicking the mouse on a command button, trigger the analyst's code. For example, clicking the folder icon associated with the source input will open a dialogue to select an ArcInfo feature class to be used as the source. Within the code for the extension, an instance of the Excel application is created and passed the U_s and a_{cs} values directly (**bold text** is processed by Excel, normal text is processed by ArcInfo, comments are in *italics*).

```
'Pass Excel the U column vector (vector of known source values)
While Not pFeat Is Nothing
  strRow=GetRow(pFeat.Value(0)) 'GetRow is a user-defined
  function
  strCol='A'
  Sheets(''Sheet2'').Select 'Store vector on Sheet 2 in
  Column A
  Range(strCol & strRow).Value=pFeat.Value(intSourceIndex)
  Set pFeat=pCursor.NextFeature
Wend
```

After all necessary data have been added to the Excel spreadsheet, Excel performs the regression (LINEST) or matrix inversion and matrix multiply (MINVERSE and MMULT) and then passes back to ArcInfo the d_c values. The target zone values can then be estimated using the above formula and added to the attribute table for the target data set. Figures 4 and 5 show the attribute tables of the target dataset before and after the interpolation, respectively, illustrating that the correct fields have been added. Note that the zero values in the POPULATION field are present because of the sliver size specified by the user.

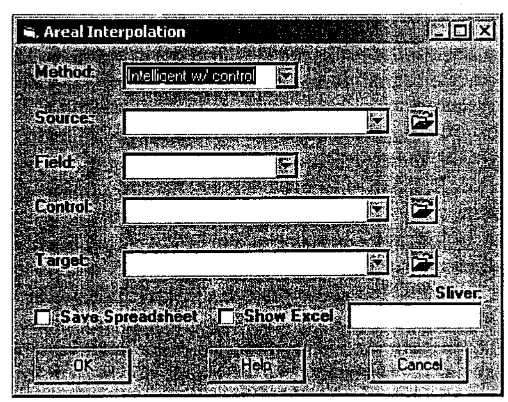

Figure 3. The Visual Basic form for the areal interpolation extension.

Attributes of target

FID	Shape	AREA	ECOREGION
0	Polygon	55938301952	NORTHWESTERN CALIFORNIA
1	Polygon	20680300544	CASCADE RANGES
2	Polygon	22858899456	MODOC PLATEAU
3	Polygon	63132901376	SIERRA NEVADA
4	Polygon	58629500928	GREAT CENTRAL VALLEY
5	Polygon	13128399872	EAST OF SIERRA NEVADA
6	Polygon	2732179968	CENTRAL WESTERN CALIFORNIA
7	Polygon	34363498496	CENTRAL WESTERN CALIFORNIA
8	Polygon	367422	CENTRAL WESTERN CALIFORNIA
9	Polygon	3086810	CENTRAL WESTERN CALIFORNIA
10	Polygon	2390680	CENTRAL WESTERN CALIFORNIA
11	Polygon	91956	CENTRAL WESTERN CALIFORNIA
12	Polygon	20802600	CENTRAL WESTERN CALIFORNIA
13	Polygon	273642	CENTRAL WESTERN CALIFORNIA
14	Polygon	30399	CENTRAL WESTERN CALIFORNIA
15	Polygon	73982099456	MOJAVE DESERT
16	Polygon	185378000	CENTRAL WESTERN CALIFORNIA
17	Polygon	33824399360	SOUTHWESTERN CALIFORNIA
18	Polygon	29188499456	SONORAN DESERT
19	Polygon	8600740	SOUTHWESTERN CALIFORNIA

Record: 0 Show: All Selected Records (0 out of 20 Sele

Figure 4. Target attribute table prior to areal interpolation. The area field values are in square metres.

5. Discussion

The above close coupling methods can be used to perform many other types of analysis using ArcInfo as the geospatial database manager and Excel as the statistical engine. Excel has many standard statistical functions which could be used for this purpose, including but not limited to: normal and Poisson distributions, *t*-test, ANOVA, Chi-square, covariance, correlation, confidence interval, *z*-test, *F*-test and Fourier analysis. Of course, these functions are not suitable for spatial data which do not meet the assumptions of the particular statistical test. If the data are autocorrelated, the standard assumption of independence in residuals is violated and some of the Excel routines may be inappropriate. Analysts using COM components from generic statistical analysis software must be aware of both the assumptions made by the statistical functions and the characteristics of the data to which the functions are applied. Because of this, a potential disadvantage of this new coupling strategy is inaccurate analysis by GIS users who are unaware of the foundations of spatial data analysis but are suddenly given access to sophisticated tools which perform complicated analyses through a graphical user interface. The only possible solution to this problem is to provide clear and succinct help files and well-documented code.

5.1. *Building on the COM framework*

The example presented here of a coupling strategy based on COM components is for a very specific scenario, using proprietary GIS and spreadsheet software and

Attributes of target

FID	Shape	AREA	ECOREGION	POPULATION
0	Polygon	55938301952	NORTHWESTERN CALIFORNIA	61638
1	Polygon	20680300544	CASCADE RANGES	10250
2	Polygon	22858899456	MODOC PLATEAU	11330
3	Polygon	63132901376	SIERRA NEVADA	246508
4	Polygon	58629500928	GREAT CENTRAL VALLEY	3201624
5	Polygon	13128399872	EAST OF SIERRA NEVADA	6505
6	Polygon	2732179968	CENTRAL WESTERN CALIFORNIA	991569
7	Polygon	34363498496	CENTRAL WESTERN CALIFORNIA	5326411
8	Polygon	367422	CENTRAL WESTERN CALIFORNIA	0
9	Polygon	3086810	CENTRAL WESTERN CALIFORNIA	0
10	Polygon	2390680	CENTRAL WESTERN CALIFORNIA	0
11	Polygon	91956	CENTRAL WESTERN CALIFORNIA	0
12	Polygon	20802600	CENTRAL WESTERN CALIFORNIA	9773
13	Polygon	273642	CENTRAL WESTERN CALIFORNIA	0
14	Polygon	30399	CENTRAL WESTERN CALIFORNIA	0
15	Polygon	73982099456	MOJAVE DESERT	2099548
16	Polygon	185378000	CENTRAL WESTERN CALIFORNIA	92
17	Polygon	33824399360	SOUTHWESTERN CALIFORNIA	14923243
18	Polygon	29188499456	SONORAN DESERT	2078374
19	Polygon	8600740	SOUTHWESTERN CALIFORNIA	0

Record: 1 Show: All Selected Records (0 out of 20 Selected.) Op

Figure 5. Target attribute table after areal interpolation. The area field values are in square metres. Features that are smaller than the sliver polygon size (10 000 000 m^2 in this example) are given a value of zero and excluded from the area weights matrix.

a programming language which is specific to a narrow range of operating systems. While this scenario may be applicable for many GIS analysts, there are many to whom it does not apply. However, the concept of using component-based software is both platform- and software-independent, and could apply to any GIS computing environment. One of the most interesting things about COM-based software for the GIS and spatial analysis community is that routines which have already been written by researchers in a COM-compliant language, such as C++, can be given a COM 'wrapper' and thus be made into COM components themselves, which would then allow them to be used in any GIS which supports COM customization.

Other researchers have recognized that advances in object-oriented programming techniques have created the potential for libraries of reusable spatial analysis software (Anselin 2000). The GIS and spatial analysis community could begin to collect a set of (possibly open-source) COM-based libraries of general spatial analysis tools that can be accessed by researchers around the globe and used from within any COM-compliant GIS. An example of such a component which could be created is the component class ArealInterpolation, exposing the IArealInterpolation interface with several methods to perform the interpolation, which could used by analysts needing to perform areal interpolation in any COM-compliant programming language (comments are in *italics*):

```
'Visual Basic example of using a proposed COM object
Dim dblTarget() as Double 'A dynamic array to store return
values
Dim pInterpOp as IArealInterpolation
Set pInterpOp=New ArealInterpolation
dblTarget=pInterpOp.InterpolateNoControl (dblSource(), _
  dblWeightsMatrix())
```

6. Conclusion

A new strategy of close-coupling between spatial analysis and GIS using the COM components of the spatial analysis software and the GIS in one code module has been presented, and an example of implementing this strategy was given using the case of performing areal interpolation. There are potentially many other ways to exploit this link, and it is possible to develop a suite of software tools (as a library of COM components) which provide these techniques to many analysts, both within geography and also in other disciplines which may rely on GIS for its spatial analysis tools (e.g. ecology, risk management and analysis). The ready availability of such techniques would further position GIS as a tool for scientific inquiry and analysis, rather than simply a sophisticated data display and query engine.

References

Anselin, L., 2000, Computing environments for spatial data analysis. *Journal of Geographical Systems*, **2**, 201–219.

Bailey, T. C., 1998, Review of statistical spatial analysis in GIS. In *Spatial Analysis and GIS*, edited by S. Fotheringham and P. Rogerson (Philadelphia: Taylor & Francis).

Bailey, T. C., and Gatrell, A. C., 1995, *Interactive Spatial Data Analysis* (Harlow, England: Longman).

Bao, B., Anselin, L., Martin, D., and Stralberg, D., 2000, Seamless integration of spatial statistics and GIS: The S-PLUS for ArcView and the S+ Grassland links. *Journal of Geographical Systems*, **2**, 287–306.

Batty, M., and Xie, Y., 1994a, Modelling inside GIS: Part 1. Model structures, exploratory spatial analysis and aggregation. *International Journal of Geographical Information Systems*, **8**, 291–307.

Batty, M., and Xie, Y., 1994b, Modelling inside GIS: Part 2. Selecting and calibrating urban models using Arc-Info. *International Journal of Geographical Information Systems*, **8**, 291–307.

Ding, Y., and Fotheringham, A. S., 1991, The integration of spatial analysis and GIS. NCGIA working paper, SUNY Buffalo, Buffalo, NY.

ESRI, 1999, *ArcInfo Online User's Guide*.

Flowerdew, R., and Green, M., 1998, Areal interpolation and types of data. In *Spatial Analysis and GIS*, edited by S. Fotheringham and P. Rogerson (Philadelphia: Taylor & Francis).

Fotheringham, S., and Rogerson, P., 1998, *Spatial Analysis and GIS*. (Philadelphia: Taylor & Francis).

Getis, A., 2000, Spatial analysis and GIS: an introduction. *Journal of Geographical Systems*, **2**, 1–3.

Goodchild, M. F., and Lam, N. S., 1980., Areal interpolation: a variant of the traditional spatial problem. *Geo-processing*, **1**, 297–312.

Goodchild, M. F., Haining, R., Wise, S., *et al.* 1992, Integrating GIS and spatial data analysis: problems and possibilities. *International Journal of Geographical Information Systems*, **6**, 407–423.

Goodchild, M. F., Anselin, L., and Deichmann, U., 1993, A framework for the areal interpolation of socioeconomic data. *Environment and Planning A*, **25**, 383–397.

Kehris, E., 1990a, Spatial autocorrelation statistics in Arc/Info. Research Report 16, North West Regional Research Laboratory, Lancaster University, Lancaster, UK.

Kehris, E., 1990b, A geographical modelling environment built around Arc/Info. Research Report 13, North West Regional Research Laboratory, Lancaster University. Lancaster, UK.

Marble, D., 2000, Some thoughts on the integration of spatial analysis and Geographic Information Systems. *Journal of Geographical Systems*, **2**, 31–35.

Microsoft, 2000, *Microsoft Developer Network online documentation* (MSDN Library July 2000).

Park, S., and Wagner, D., 1997, Incorporating CA simulators as analytical engines in GIS. *Transactions in GIS*, **2**, 213–231.

Pebesma, E., and Wesseling, C. G., 1998, Gstat: a program for geostatistical modelling, prediction and simulation. *Computers & Geosciences*, **24**, 17–31.

Rowlingson, B. S., Flowerdew, R., and Gatrell, A., 1991, Statistical spatial analysis in a geographical information systems framework. North West Regional Research Laboratory, Lancaster University. Lancaster, UK.

Sadahiro, Y., 1999, Accuracy of areal interpolation: a comparison of alternative methods. *Journal of Geographical Systems*, **1**, 323–346.

Sadahiro, Y., 2000, Accuracy of count data transferred through the areal weighting interpolation method. *International Journal of Geographical Information Science*, **14**, 25–50.

Zhang, Z., and Griffith, D., 2000, Integrating GIS components and spatial statistical analysis in DBMS's. *International Journal of Geographical Information Science*, **14**, 543–566.