Visual Basic time-space query modeling for geovisualization of life paths

Ryan Heaslip

West Virginia University

Follow this and additional works at: https://researchrepository.wvu.edu/etd

Recommended Citation
Heaslip, Ryan, "Visual Basic time-space query modeling for geovisualization of life paths" (2004). Graduate Theses, Dissertations, and Problem Reports. 1543.
https://researchrepository.wvu.edu/etd/1543

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.
Visual Basic Time-Space Query Modeling for Geovisualization of Life Paths

Ryan Heaslip

Problem Report submitted to the Eberly College of Arts and Science at West Virginia University in partial fulfillment of the requirements for the degree of Master of Arts in Geography

Gregory Elmes, Ph.D., Chair
Ge Lin, Ph.D.
Jennifer Miller, Ph.D.

Department of Geography

Morgantown, West Virginia 2004

Keywords: Geography, GIS, Visual Basic, Time Space Aquarium, Life Path, 3D Geovisualization, Time Geography, Medical Geography

Copyright 2004 Ryan A. Heaslip
Abstract

Visual Basic Time-Space Query Modeling For Geovisualization of Life Paths

Ryan A Heaslip

The origins and incidence of many health problems are related to geographical circumstances. To be able understanding the underlying contributing factors requires conceptualizing the problem in geographical terms. A temporal dataset of individuals’ locations was acquired from an annual census taken in Sweden. A space-time query model was constructed to explore associations between an individual’s residential locations for given periods of time to those of other individuals with similar health, or health related factors. The space-time query model was created in the Visual Basic extension of ArcGIS 8.3. Results were in the form of selections that could be exported into a 3D space-time aquarium for geovisualization in ArcScene. This query model has the potential to be a very powerful tool with which the possible causal factors of many health-related issues can be explored, as well as the prediction of future health issues based on exposures to pathogens at given times and places.
# Table of Contents

Title Page ................................................................. i
Abstract ........................................................................ ii
Table of Contents ............................................................. iii
Introduction ...................................................................... 1
Methods ........................................................................... 13
Expected Results ............................................................ 15
Conclusion ........................................................................ 15
Appendix A: Results of Example Queries ...................... 17
Appendix B: Visual Basic Interface .............................. 27
Appendix C: Visual Basic Code .................................... 28
Appendix D: Attributes of Excel Data Table .................. 39
References ........................................................................ 40
Introduction

Many of the concepts concerning the causes, outcomes, and distribution of health and disease have been formed on the basis of medical and biological knowledge. Since the origins and incidence of many health problems are related to geographical circumstances, understanding the underlying contributing factors requires conceptualizing the problem in geographical terms (Schærström 1996). Medical geography seeks to employ the concepts and methods used in the field of geography to understand and examine health-related issues (Meade & Earickson 2000). Its foundation is based upon the understanding that the spatial distribution of diseases and proximity to pathogenic influences is important. Through mapping the spatial occurrence of disease one can investigate and hypothesize about their causes (Schærström 1996). Knowing where and when a disease occurs and where it does not occur provides insight into the factors that promote, cause or counteract the disease. Previously both epidemiological and geographical studies of disease occurrence have been criticized for being not completely integrated across scale and such studies have often encountered problems caused by the enforced aggregation of data from the individual level (Aase 1989, Bentham 1988, and Hägerstrand 1969). In addition to questions of scale, to obtain a more accurate representation of space and a more powerful analytical method, time needs to be incorporated into medical geography (Schærström 1996). More specifically, we need to consider concept of space-time and its representation in GIS.

Time geography seeks to analyze the consequences of the coexistence of phenomena existing in time and space. It examines the effects of different phenomena and processes existing side-by-side, for example relating climate changes to the
movement of tree species on the North American continent, over the past 10,000 years. Many of the current implementations of Geographic Information Systems (GIS have not yet modeled time-geography fully. The ability of a GIS, to display historical information, produce forecasts and track changes over time is a relatively new concept (Langran 1992). To know how to display temporal information in a GIS one must first have a working knowledge of the concepts of time itself.

In early cultures a cyclical view of time was commonly held. This was a reflection of cycles of nature and the rhythms of everyday life. The early Greeks associated time on a daily scale consisting of the ordered rhythm of human activities within the seasons of the year, and the repeating cycles of events, such as bird migrations and planting time. The Mayan culture believed that time would repeat itself every 260 years (Peuquet 2002). Our culture commonly holds a linear view of time, as a line without endpoints; an endless line that stretches into the infinity of the past and future. Two questions arise: How can this time line be represented? And, What if any are the significant components of it that should be illustrated? (Langran 1992).

The answers to these questions lie in that of ‘cartographic time’. Cartographic time “distills the characteristics of time that are essential for representing spatio-temporality in the most pragmatic and generic fashion possible” (Langran 1992 p 28). The very essence of cartographic time lies within temporal boundaries. Much like spatial boundaries that are formed when adjacent locations differ, temporal boundaries are formed when the adjacent states of the modeled system differ from the previously recorded state. In essence, Figure 1 illustrates that when a current state is being observed,
a temporal boundary is formed at the moment in time an event occurs to change the current state, (Langran 1992).

![Diagram of cartographic time]

Figure 1: The topology of cartographic time. Relationships of objects to maps, mutations to events, and version to states. 

Peuquet (1994) notes that there are three ways currently used to represent time and temporal boundaries cartographically, these are: by displaying the sequence of time in a number of ‘still shots’ (time-slices), by dynamically modifying the display element via programmatic or interactive control; or by using a static map display with supplementary graphic or charts depicting the change in a specific variable. The most heavily used method of establishing temporal boundaries in GIS is through that of time slices. Time slices are snap shots of dynamic situations, such as diffusion, movement, or migration. In time slices, time is essentially fossilized and made static in fact time is eliminated. If certain incidents occur between slices, they are not recorded, nor can they be verified. These portions of time do not offer any type of continuous data (Langran 1992).
So far, time geography has rarely been applied to analyze how place might affect the health of those who spend their time there. Would it be possible to use time geography to analyze how accessible (or exposed) people are to pathogenic agents and process? Schærström (1996) would answer positively. If all people’s life paths were mapped along with their current health status, correlations between people’s health problems or illness and their proximity to others with similar problems could be constructed and shown. The main basis of thinking in time geography with health is that the “individual is the smallest indivisible unit in social contexts, and that the locations and movements of individuals can be followed and visualized as continuous paths in the spatial and temporal dimensions” (Schærström 1996 p 97). In this sense time geography can be used as a tool to enhance a map with a temporal dimension that a person can easily visualize, analyze, and process the movements of individuals in space or individuals’ life paths (See figure 2).
Hägerstrand (1970) was the first to formulate a simple concept of representing the
temporal geographic movement and actives of an individual in a space-time ‘prism’
defined by a set of constraints. This method conceived by Hägerstrand was modified by
Kwan to show 3D geovisualization of space-time activity patterns (Kwan 2000). Kwan
used the geographical X and Y coordinates to show the location(s) of an individual’s
activity and the Z value to represent time. From these geovisualizations Kwan was able
to create activity density patterns in geographic space at a fine granularity spatial and
temporal resolution within the city (See Figure 3). The maps had a single linear display
of how long the person was engaged in the activity at a specific location. These maps
were effective in displaying the concentrations of locations of participants performing their daily activities, such as workplaces, schools, shopping, etc. On the other hand the maps did not display the actual paths of the individual through the city throughout the day (Kwan & Lee 2004).

In addition to constructing the geovisualization of activity density patterns Kwan (2000) also used GIS to construct a ‘3D space-time aquarium’ originally conceived by Hägerstrand (1970). In the space-time aquarium, the vertical axis is represented by the time of day and the horizontal plane represents the spatial scope of the study area. Individual space-time paths are portrayed as trajectories in this 3D aquarium. Unlike the activity pattern maps that showed where an individual was doing a certain activity for a certain amount of time, the 3D space-time aquarium showed the activity and path of an individual throughout the day. This type of geovisualization allows for the easy visual

Figure 3. Simple activity patterns in space-time. (Kwan & Lee 2004)
interpretation of where an individual has been in a day as well as the other individuals that they have been in contact with (Kwan 2000).

Figure 4. Space-time aquarium showing the space-time paths of African and Asian Americans in Portland, Oregon. (Kwan & Lee 2004)

The space-time aquarium that Kwan constructed is a significant method of representing and visualizing the movements of individuals’ The spatial relationships one to another, such as collocation in time and space, could then be linked to environment and health information.

In 2003, Elmes acquired a time and space referenced data set for a large sample of individuals in two counties in Sweden from the School of Public Health, of the Medical University of Linköping. The data was collected on a yearly census from the time period
of 1973 to 2000. An individual’s residential location at the time of census was recorded for each year, as a latitude and longitude coordinate to within a kilometer; in addition the individual’s UN (Unique Number – assigned to every Swedish resident) was recorded along with their sex and year of birth. This data set was placed in a Microsoft Excel table and organized to have the following fields of information: A UN field, which can be used as a primary key to identify an individual. A Year Born field identifies when an individual was born, a Sex field displays the sex of the individual, and finally fields containing the latitude and longitude (X and Y coordinates) of each individual’s residential location for every year, from 1973-2000. A total of 58,071 life paths exist in this data set, from which there are 921,624 individual locations. (See Table 1.)

The three main questions being asked using this data as a basis are:

1) How can space-time data be represented in a GIS environment?

2) How are space-time queries performed upon that data?

3) How may the query results be visualized using GIS?

The representation of the Swedish space-time data in a GIS environment and the ability to perform space-time queries upon that data can be a great asset to exploration and analysis of the geography of health. Since this data set shows the long-term residential movement and migration of individuals within Sweden, health registry data, also indexed by the unique number, pertaining to these individuals can be linked to their geographic trajectories. Similarly, environmental data can be georegistered, permitting association between place and length of residence with exposure to possible environmental pathogens. Associations between the individuals’ health and that of other individuals could be analyzed using both spatial and temporal queries. These correlations
would be extremely useful in identifying geographic areas or regions, which are prone to
disease or other chronic illnesses.

In addition, a 3D geovisualization of the individuals’ locations and trajectories
could be constructed in a space-time aquarium. Three-dimensional mapping can be a
very powerful tool to visualize the results of queries being conducted and to explore
potential hotspots of disease.

To conceptualize this model, the migration data must be incorporated in a GIS
and a query structure must be developed. The tabular location data must first be entered
into a GIS and transformed into geometric objects as shapefiles or layers. Since the
Geography Department at West Virginia University has invested large amounts of time,
money, and knowledge into ESRI’s ArcGIS and ArcGIS has a Microsoft Visual Basic
(VB) extension in which queries can be constructed and assimilated into the software, the
ArcInfo software package was chosen to construct the model.

As with any information system this model must have the ability to represent the
entities of interest in order to answer questions about them. To accomplish this goal, the
spatiotemporal queries in the model have to relate to three key aspects: what, where, and
when, in context to information about properties or attributes, location or spatiality, time
instance or duration, and transitions or development. In answering these aspects four
classes of spatiotemporal queries can be formed: (1) queries about attributes; (2) queries
about locations, spatial properties or spatial relationships; (3) queries about time,
temporal properties, and temporal relationships; (4) queries about spatiotemporal
behavior and relationships (Yuan and McIntosh 2002).
Spatiotemporal queries can involve search in both spatial and temporal dimensions. The queries can be space dominant; i.e. examining the locational context without regard for time; temporal dominant; examining the temporal context without regard for location; or an incorporation of the two, using both the locational and temporal attributes to identify a feature. (Langran 1993). Table 1 gives example to the dimensional dominance of spatiotemporal queries.

Table 1. The dimensional dominance of sample spatiotemporal queries (Langran 1993)

<table>
<thead>
<tr>
<th>Query</th>
<th>Space dominant</th>
<th>Time dominant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieve a snapshot of an area at a given time</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Retrieve snapshot(s) of an area at even intervals during a time span</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Retrieve all feature that have ever held certain attributes</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Retrieve snapshot(s) of an area at the moments(s) when a feature holds a certain attributes</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Retrieve the history of a given feature</td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

Since most GIS software packages already provide a built in query by attribute the first aspect of spatiotemporal queries is not addressed specifically in the time-space query model. The model does enable spatial queries, temporal queries, and space-time queries. Figure 5 depicts the incorporation of the spatial, temporal, and space-time queries, in the Visual Basic interface along with the exportation and 3-dimensional rendering of the data in ArcScene. The space queries will be concerned with the locations of individuals in reference to other individuals or events that may be occurring, whereas the temporal queries will address the locations of individuals during specific time slices. The incorporation of both the temporal and the spatial queries (space-time
queries) will solicit individuals’ locations based on both their relative location to other individuals or events, but also to the time periods that those individuals resided at those locations.

Figure 5: Structure of Time-Space Query Model.

Using the VB-based query interface, the user will select and input the queries that they wish to perform. At runtime, the VB script will access the space-time data being stored in point and polyline format, and apply the appropriate queries to the data. The data will be exported in the form of a map that will have the locations and life paths
meeting the requirements of the query selected. The user would then have the option of outputting those results into a shapefile that could easily be opened and rendered in the ArcScene environment.

The space-time queries to be constructed and run in a VB environment are as follows:

1. All individuals’ location(s) during a specific time. This is a classic spatial query focusing on who existed during the defined time period.
2. A specific individual’s location during a specific time. This is a spatial query focusing on the individual that existed during the defined time period.
3. All individuals’ locations within a given distance of a specific point. This is a spatial query focusing on: throughout all time periods the individuals that where within a certain distance of a point.
4. A specific individual’s location(s) within a given distance of a specific point. This is a spatial query focusing on: throughout all time periods the individual that was within a certain distance of a point.
5. All individuals’ locations within a given distance of a specific individual’s location(s). This is a spatial query focusing on: throughout all time periods the individuals that where within a certain distance of a certain individual.
6. A specific individual’s location(s) within a given distance of a specific individual’s location(s). This is a spatial query focusing on: throughout all time periods a certain individual that was within a certain distance of a certain individual.
7. A specific individual’s location(s) within a specific temporal domain from a specific point. This is a space-time query, which focuses on not only the distance a specific individual is from a point but also the user defined temporal period that they resided at those locations.
8. All individuals’ location(s) within a specific temporal domain from a specific point. This is another space-time query which focuses on the distance specific individuals are from a point and the user defined temporal period that they resided at those locations.
9. A specific individual’s location(s) within an area and the time period of an event happening. This is a space-time query based on a point in polygon focusing on the locations of a specific individual in contact with the specified event.
10. All individuals’ location(s) within an area and the time period of an event happening. Once again this is a space-time query focusing on the locations of all individuals in contact with a specific event.
Methods

Excel files of the raw data were imported into ArcGIS 8.3 and latitude and longitude coordinates were made into points using the Add XY data tool in ArcMap. All the other fields were input as a point attribute table (PAT).

The PAT attribute fields were established as follows:

- **FID**: A Primary Key ID assigned to each point by ArcGIS. Used to distinguish each point from one another.
- **Shape**: An attribute assigned by ArcGIS to identify the object as a point line or polygon.
- **UN**: The unique number of the individual. Can be used to identify all the locations that an individual has resided. No more than one of these exist for each year.
- **BORN.YEAR**: The year in which the individual was born.
- **Year**: The temporal year in which the individual existed at the specific location.

![Attributes of All_People](image)

Figure 5: The attribute table of the data.

The life paths were created in ArcView 3.3 using an Avenue™-based point to polyline script. An Avenue™ script was used to transfer attributes from a Point theme to a Line theme of Year<sub>t</sub> - Year<sub>t+n</sub> from the point data to the life paths, so that polylines could be constructed in 3D space. The life paths were created as a visual reference to link all the locations that an individual has resided. Since these life paths were created using the location of addresses where an individual resides for a given time period, the polylines should not be interpreted as actual space-time movements. To obtain actual space-time movements of the individuals, a finer temporal scale, or granularity, of their movements in space would have to be obtained.
The query application will consist of a series of VB forms allowing the user to select from menus the spatial, temporal, or space-time queries the user wishes to perform.

Since the Swedish data set is at a temporal scale (or granularity) of annual time slices, it is not possible to represent where the individuals had visited or traveled during that period; the dataset only consists of the residential locations of the individuals, not where they worked or where they were on a daily or monthly basis. In the model it will have to be assumed that individuals in close spatial and temporal contact with one another potentially experienced close contact in reality.

Another limitation is that of the vast size of the data set. With current hardware and software RAM limitations, it is impossible to display all 921,624 individuals’ locations and 58,071 life paths in a 3D space-time aquarium. Small subsets of the data have been extracted to demonstrate the concepts, which have been implemented. Hence if a query is performed on a full dataset in which a very large number of individuals’ life paths are selected, the current computational ability of ArcScene to display these in a 3D space-time aquarium takes an excessively long time or will cause the computer to crash.

A final limitation is that the data exists for the time period of 1973-2000 for the geographic region of two counties in Sweden (Östergötland and Kalmar). This is important to note, since individuals that left the study area (either by moving or by death) during that time period disappear from the map. Any individual that moved into the study area appears suddenly, in temporal space, on the map. In a few rare cases, there are actually individuals who moved out of the study area, for a few years then moved back into the study area. This makes their life paths broken, in which there are a series of years that the locations of the individuals were not recorded.
**Expected Results**

Once the user selects the query and has inputted the spatial or temporal variables to the GUI, an ArcMap project will appear showing the selected points, and life-paths associated with those points. From this ArcMap display, the user could export the points into a separate shapefile.

The data could be viewed in a 3D space-time aquarium using ArcScene. The attributes already exist in the data to view it in the third dimension; the user need only apply the year fields to the base heights of the individuals’ point locations and the life-path polylines.

**Conclusion**

The rationale behind the development of these space-time queries and 3D space-time aquariums are to allow the user to assimilate or relate the space-time paths with environmental and health data. The user would then be able to find and develop spatial and temporal associations between people and their health. These queries have the potential to become a very powerful tool through which the potential causes of many health issues can be explored, as well as the prediction of future health outcomes.

Due to short-fallings in the implementation of the VB code, a fully functional space-time query interface could not be achieved. In particular this was in the form of not implementing the MakePermanent command listed under ISelectionSet.

In the future development of the model the construction of a fully functional query interface is needed which will allow the user to have full use of all the queries. A better way of memory usage in code needs to be found, so that the interface can function with large data sets. Finally the model needs to be incorporated into ArcGIS 9
and AML coding to see if there is better functionality than that of Visual Basic.
APPENDIX A: Results of Example Queries

Query 1: All individuals’ location(s) during a specific time.
Individuals’ locations in 1988
Query 2: A specific individual’s location during a specific time.
Individual UN 86’s location to 1988

Legend
- Selected Individual Life-Path
- Unselected Individual Life-Path
- Sweden
Query 3: All individuals’ locations within a given distance of a specific point.
All individuals within 6 miles of a user defined point

Legend
- Selected Individual Life-Path
- Unselected Individual Life-Path
- User Defined Point
- Buffer of Point
- Sweden
Query 4: A specific individual’s location(s) within a given distance of a specific point. Individual UN 86 within 6 miles of a user defined point
Query 5: All individuals’ locations within a given distance of a specific individual’s location(s).
All individuals within 12 miles of individual UN 67.
Query 6: A specific individual’s location(s) within a given distance of a specific individual’s location(s).
Individual UN 153 within 12 miles of individual UN 67
Query 7: A specific individual’s location(s) within a specific temporal domain from a specific point.
Individual UN 230 within 6 mile of user defined point between the years of 1980 and 1993
Query 8: All individuals’ location(s) within a specific temporal domain from a specific point.
All individuals within $3\frac{1}{2}$ miles of user defined point between the years of 1982 and 1990
Query 9: A specific individual’s location(s) within an area and the temporal time of an event occurring.

Individual UN 153 in contact with event

Legend
- Selected Individual Life-Path
- Unselected Individual Life-Path
- Event Occurring
- Sweden
Query 10: All individuals’ location(s) within an area and the temporal time of an event occurring.
All individuals in contact with event
APPENDIX B: Visual Basic Interface

Figure 1. Visual Basic Interface
APPENDIX C: Visual Basic Code

Query 1 All individuals’ location(s) during a specific time.

Option Explicit

Private Sub CmdBack_Click()
    Unload FrmMultipleTime
    FrmMultiple.Show
End Sub

Private Sub cmdFinish_Click()
    Dim StartT As Integer

    Dim pMxDoc As IMxDocument
    Dim pMap As IMap
    Dim pActiveView As IActiveView
    Dim pFeatureLayer As IFeatureLayer
    Dim pFeatureSelection As IFeatureSelection
    Dim pQueryFilter As IQueryFilter

    Set pMxDoc = Application.Document
    Set pMap = pMxDoc.FocusMap
    Set pActiveView = pMap
    StartT = txtStart_Time.Value
    'MsgBox StartT

    'For simplicity sake let's use the first layer in the map
    If Not TypeOf pMap.Layer(0) Is IFeatureLayer Then Exit Sub
    Set pFeatureLayer = pMap.Layer(0)
    Set pFeatureSelection = pFeatureLayer 'QI

    'Create the query filter
    Set pQueryFilter = New QueryFilter
    pQueryFilter.WhereClause = "Year = " & StartT

    'Invalidate only the selection cache
    'Flag the original selection
    pActiveView.PartialRefresh esriViewGeoSelection, Nothing, Nothing
    'Perform the selection
    pFeatureSelection.SelectFeatures pQueryFilter, esriSelectionResultNew, False
    'Flag the new selection
    pActiveView.PartialRefresh esriViewGeoSelection, Nothing, Nothing
End Sub

Query 2 A specific individual’s location during a specific time.
Option Explicit

Private Sub CmdBack_Click()
    Unload FrmSingleTime
    FrmSingle.Show
End Sub

Private Sub cmdFinish_Click()
    Dim StartT As Integer
    Dim pMxDoc As IMxDocument
    Dim pMap As IMap
    Dim pActiveView As IActiveView
    Dim pFeatureLayer As IFeatureLayer
    Dim pFeatureSelection As IFeatureSelection
    Dim pQueryFilterYear As IQueryFilter
    Dim pQueryFilterUN As IQueryFilter

    Set pMxDoc = Application.Document
    Set pMap = pMxDoc.FocusMap
    Set pActiveView = pMap
    StartT = txtStart_Time.Value
    'MsgBox StartT

    'For simplicity sake let's use the first layer in the map
    If Not TypeOf pMap.Layer(0) Is IFeatureLayer Then Exit Sub
    Set pFeatureLayer = pMap.Layer(0)
    Set pFeatureSelection = pFeatureLayer 'QI

    'Create the query filter
    Set pQueryFilterYear = New QueryFilter
    Set pQueryFilterUN = New QueryFilter
    pQueryFilterYear.WhereClause = "Year = " & StartT
    pQueryFilterUN.WhereClause = "UN = " & UNValue

    'Invalidate only the selection cache
    'Flag the original selection
    pActiveView.PartialRefresh esriViewGeoSelection, Nothing, Nothing
    'Perform the selection
    pFeatureSelection.SelectFeatures pQueryFilterYear, esriSelectionResultNew, False
pFeatureSelection.SelectFeatures pQueryFilterUN, esriSelectionResultAnd, False
'Flag the new selection
pActiveView.PartialRefresh esriViewGeoSelection, Nothing, Nothing

End Sub

Query 5: All individuals’ locations within a given distance of a specific individual’s location(s).

Option Explicit
Dim pApp As IApplication

Private Sub CmdBack_Click()
    Unload FrmMultipleDistIndPerson
    FrmMultiple.Show
End Sub

Private Sub cmdSet_Click()
    Call Select_the_points
    Call Create_buffer_around_selected_points
    'Call Select_all_the_buffers
    'Call Select_Points_In_Selected_Poly

    Unload FrmMultipleDistIndPerson
End Sub

Private Sub Select_the_points()

'Selecting the Points to be buffered

    Dim StartT As Integer
    Dim EndT As Integer
    Dim UNValue2 As Integer
    Dim pMxDoc As IMxDocument
    Dim pMap As IMap
    Dim pActiveView As IActiveView
    Dim pFeatureLayer As IFeatureLayer
    Dim pFeatureSelection As IFeatureSelection
    Dim pQueryFilterStartT As IQueryFilter
    Dim pQueryFilterEndT As IQueryFilter
    Dim pQueryFilterUN As IQueryFilter
Set pMxDoc = Application.Document

Set pApp = New AppRef

Set pMap = pMxDoc.FocusMap
Set pActiveView = pMap
StartT = txtStartTime.Value
EndT = txtEndTime.Value
UNValue2 = TxtUNValue.Value
'Message StartT
'clear the Graphics

Dim pGC As IGraphicsContainer
Set pGC = pMap
pGC.DeleteAllElements

'For simplicity sake let's use the first layer in the map
If Not TypeOf pMap.Layer(1) Is IFeatureLayer Then Exit Sub
Set pFeatureLayer = GetFeatureLayer("Sample_set", pApp)
Set pFeatureSelection = pFeatureLayer 'QI

'Create the query filter
Set pQueryFilterStartT = New QueryFilter
Set pQueryFilterEndT = New QueryFilter
Set pQueryFilterUN = New QueryFilter
pQueryFilterStartT.WhereClause = "Year >= " & StartT
pQueryFilterStartT.WhereClause = "Year <= " & EndT
pQueryFilterUN.WhereClause = "UN = " & UNValue2

'Invalidate only the selection cache
'Flag the original selection
pActiveView.PartialRefresh esriViewGeoSelection, Nothing, Nothing
'Perform the selection
pFeatureSelection.SelectFeatures pQueryFilterUN, esriSelectionResultNew, False
pFeatureSelection.SelectFeatures pQueryFilterStartT, esriSelectionResultAnd, False
pFeatureSelection.SelectFeatures pQueryFilterEndT, esriSelectionResultAnd, False
'Flag the new selection
pActiveView.PartialRefresh esriViewGeoSelection, Nothing, Nothing
End Sub

''**********************************************************************
Private Sub Create_buffer_around_selected_points()

Const strFolder As String = "D:\Ryan_work"
'Const strName As String = "StorePoly" ' Don't include .shp extension
Const strShapeFieldName As String = "Shape"

Page 31
' Open the folder to contain the shapefile as a workspace
Dim pFWS As IFeatureWorkspace
Dim pWorkspaceFactory As IWorkspaceFactory
Set pWorkspaceFactory = New ShapefileWorkspaceFactory
Set pFWS = pWorkspaceFactory.OpenFromFile(strFolder, 0)

"Running the Buffer
,'
Dim pMxDoc As IMxDocument
Dim pActiveView As IActiveView
Dim pGraphicsContainer As IGraphicsContainer
Dim pEnumFeature As IEnumFeature
Dim pFeature As IFeature
Dim pTopoOp As ITopologicalOperator
Dim pElement As IElement
Dim strBufferDistance As Double
Dim MetersToKM As Double
Dim pGeometrybag As IGeometryCollection
Dim pPolygon As IPolygon
Dim pPolygon2 As IFeature
Dim m_pBufferPolygon As IPolygon
Dim m_pLastBufferedExtent As IEnvelope
Dim pFeaturePoly As IFeature
Dim pFeatureClass As IFeatureClass
Dim KM As Double
Dim Meters As Double
' Set pApp = New AppRef

Dim pPolyLayer As IFeatureLayer
' Set pPolyLayer = pMxDoc.FocusMap.Layer(0)
Set pPolyLayer = GetFeatureLayer("StorePoly", pApp)
Set pMxDoc = Application.Document
Set pActiveView = pMxDoc.FocusMap
Set pGraphicsContainer = pMxDoc.FocusMap
Set pGeometrybag = New GeometryBag

' Verify there is a feature selection
If pMxDoc.FocusMap.SelectionCount = 0 Then Exit Sub

' Get a buffer distance from the user
' MsgBox MetersToKM
Meters = txtDistance.Value
KM = Meters * 1000
strBufferDistance = KM
' Buffer all the selected features by the BufferDistance
' and create a new polygon element from each result
Set pEnumFeature = pMxDoc.FocusMap.FeatureSelection
pEnumFeature.Reset
Set pFeature = pEnumFeature.Next
Do While Not pFeature Is Nothing
Set pTopoOp = pFeature.Shape
'Set pElement = New PolygonElement
Set pPolygon2 = pPolyLayer.FeatureClass.CreateFeature
Set pPolygon2.Shape = pTopoOp.Buffer(CInt(strBufferDistance))
'pElement.Geometry = pTopoOp.Buffer(CInt(strBufferDistance))
'pPolygon2.Shape = pElement.Geometry
'pGraphicsContainer.AddElement pElement, 0
'pGeometryBag.AddGeometry pPolygon2
pPolygon2.Store
'get next feature
Set pFeature = pEnumFeature.Next
Loop
' union all the buffers into one polygon
Set m_pBufferPolygon = New Polygon
Set pTopoOp = m_pBufferPolygon
Set m_pLastBufferedExtent = m_pBufferPolygon.Envelope

MsgBox "Buffer Complete", vbOKOnly

'Redraw the graphics
pActiveView.PartialRefresh esriViewGraphics, Nothing, Nothing

'Private Sub Select_all_the_buffers()
'Select all the Points inside the Buffers
' Dim pMxDoc As IMxDocument
' Dim pActiveView As IActiveView

Set pMxDoc = Application.Document
Set pActiveView = pMxDoc.FocusMap

' Get the polygon layer (this code assumes the name is Buffer_of_Point)
Dim pPolyLayer2 As IFeatureLayer
Set pPolyLayer2 = GetFeatureLayer("StorePoly", pApp)

'select all the polygons in that layer so that the points inside them can be selected
Dim pPolySelection As IFeatureSelection
Set pPolySelection = pPolyLayer2

Dim pQueryFilterPoly As IQueryFilter
Set pQueryFilterPoly = New QueryFilter
Dim PolygonText As String
PolygonText = 0
pQueryFilterPoly.WhereClause = "Id = " & PolygonText

MsgBox "The Polygons have been selected", vbOKOnly

'Invalidate only the selection cache
'Flag the original selection
pActiveView.PartialRefresh esriViewGeoSelection, Nothing, Nothing
'Perform the selection
pPolySelection.SelectFeatures pQueryFilterPoly, esriSelectionResultAdd, False
'Flag the new selection
'pActiveView.PartialRefresh esriViewGeoSelection, Nothing, Nothing
'End Sub
'
'Private Sub Select_Points_In_Selected_Poly()

MsgBox "SelectionSet done", vbOKOnly
'Private Sub Select_Points_In_Selected_Poly()

'Dim pMxDoc As IMxDocument
'Dim pAView As IActiveView
Dim pMap As IMap
Dim pFLayer_point As IFeatureLayer
Dim pFLayer_poly As IFeatureLayer
Dim pFSelection_point As IFeatureSelection
Dim pFSelection_poly As IFeatureSelection
Dim pSelectionSet As ISelectionSet
Dim pFClass_poly As IFeatureClass
Dim pFCursor As IFeatureCursor
'Dim pFeature As IFeature
Dim pGeometry As IGeometry
Dim pSpatialFilter As ISpatialFilter
Dim pFLayer As IFeatureLayer
Set pMap = pActiveView.FocusMap.FacetMap.
'Set pMxDoc = Application.Document
'Set pAView = pMxDoc.ActiveView
'Set pMap = pMxDoc.FocusMap

'Set pFLayer_point = pMap.Layer(0)
'Set pFLayer_poly = pMap.Layer(1)
    Set pFLayer_point = GetFeatureLayer("Sample_set", pApp)
    Set pFLayer_poly = GetFeatureLayer("StorePoly", pApp)
Dim pFeatureSelection As IFeatureSelection
'Set pFLayer = pMxDoc.SelectedLayer
'Set pFeatureSelection = pFLayer.FeatureClass.Select()
'Set pSelectionSet = pFeatureSelection.SelectionSet

Set pSpatialFilter = New SpatialFilter

Set pFClass_poly = pFLayer_poly.FeatureClass
'Set pFSelection_poly = pFLayer_poly
'Set pSelectionSet = pPolySelection.SelectionSet 'pFSelection_poly.SelectionSet

pSelectionSet.Search Nothing, False, pFCursor

Set pFeature = pFCursor.NextFeature
While Not pFeature Is Nothing
    Set pGeometry = pFeature.Shape
    Set pFeature = pFCursor.NextFeature
Wend

Set pSpatialFilter.Geometry = pGeometry
pSpatialFilter.GeometryField = pFClass_poly.ShapeFieldName
pSpatialFilter.SpatialRel = esriSpatialRelIntersects

Set pFSelection_point = pFLayer_point

pActiveView.PartialRefresh esriViewGeoSelection, Nothing, Nothing
pFSelection_point.SelectFeatures pSpatialFilter, esriSelectionResultAdd, False
pFSelection_point.SelectFeatures pSpatialFilter, esriSelectionResultAdd, False
pActiveView.PartialRefresh esriViewGeoSelection, Nothing, Nothing
End Sub

'End Sub
Private Sub Populate_ListBox_With_UN()
Dim pMxDoc As IMxDocument
Dim pAView As IActiveView
Dim pMap As IMap
Dim pFeatureLayer As IFeatureLayer
Dim pFeatureSelection As IFeatureSelection
Dim pSelectionSet As ISelectionSet
Dim pFeatureCursor As IFeatureCursor
Dim pFeature As IFeature
Dim lFieldIndex As Long

Set pMxDoc = Application.Document
Set pAView = pMxDoc.ActiveView
Set pMap = pMxDoc.FocusMap

'Get selection from selected layer in TOC
Set pFeatureLayer = GetFeatureLayer("Sample_set", pApp)
Set pFeatureSelection = pFeatureLayer
Set pSelectionSet = pFeatureSelection.SelectionSet

' Return a feature cursor for the selection set
pSelectionSet.Search Nothing, False, pFeatureCursor

lFieldIndex = pFeatureCursor.FindField("UN")
Set pFeature = pFeatureCursor.NextFeature

Do Until pFeature Is Nothing
    lstbxUNValues.AddItem (pFeature.Value(lFieldIndex))
    Set pFeature = pFeatureCursor.NextFeature
Loop

End Sub

Function GetFeatureLayer(sName As String, pApp2 As IApplication) As IFeatureLayer
On Error GoTo erh

Dim pMxDoc As IMxDocument
Set pMxDoc = pApp2.Document

Dim pMap As IBasicMap
Set pMap = pMxDoc.FocusMap

Dim LyrCount As Integer
LyrCount = pMap.LayerCount

Dim pLayer As ILayer
Dim sLayerName As String
Dim i As Integer

If LyrCount <> 0 Then
  For i = 0 To LyrCount - 1
    Set pLayer = pMap.Layer(i)
    If TypeOf pLayer Is IFeatureLayer Then
      sLayerName = pLayer.Name
      If (StrComp(sName, sLayerName, vbTextCompare) = 0) Then
        Set GetFeatureLayer = pLayer
        Exit Function
      End If
    End If
  Next i
Else
  Set GetFeatureLayer = Nothing
End If
Set pMap = Nothing
Set pLayer = Nothing
Set pMxDoc = Nothing
Exit Function

Private Sub CommandButton1_Click()
Dim pMxDoc As IMxDocument
Dim pActiveView As IActiveView

  Set pMxDoc = Application.Document
  Set pActiveView = pMxDoc.FocusMap
Dim pFeatureLayer As IFeatureLayer
Dim pFeatureSelection As IFeatureSelection
Dim pSelectionSet As ISelectionSet
Dim pFeatureCursor As IFeatureCursor
Dim pFeature As IFeature
Dim lFieldIndex As String

'Get selection from selected layer in TOC
Set pFeatureLayer = pMxDoc.SelectedLayer
Set pFeatureSelection = pFeatureLayer
Set pSelectionSet = pFeatureSelection.SelectionSet

' Return a feature cursor for the selection set
pSelectionSet.Search Nothing, False, pFeatureCursor

lFieldIndex = pFeatureCursor.FindField("UN2")
Set pFeature = pFeatureCursor.NextFeature

Dim pQueryFilterUN As IQueryFilter
    Set pQueryFilterUN = New QueryFilter
    Dim UNValuesInList As String
Do Until pFeature Is Nothing
    lstbxUNValues.AddItem (pFeature.Value(lFieldIndex))
    Set pFeature = pFeatureCursor.NextFeature
Loop

UNValuesInList = lstbxUNValues.Text
pQueryFilterUN.WhereClause = "UN2 = " & UNValuesInList

' Invalidate only the selection cache
' Flag the original selection
pActiveView.PartialRefresh esriViewGeoSelection, Nothing, Nothing
' Perform the selection
pFeatureSelection.SelectFeatures pQueryFilterUN, esriSelectionResultAdd, False

' Flag the new selection
pActiveView.PartialRefresh esriViewGeoSelection, Nothing, Nothing
' Loop
End Sub
<table>
<thead>
<tr>
<th>UN</th>
<th>BORN</th>
<th>SEX</th>
<th>X73</th>
<th>Y73</th>
<th>X74</th>
<th>Y74</th>
<th>X75</th>
<th>Y75</th>
</tr>
</thead>
<tbody>
<tr>
<td>X76</td>
<td>Y76</td>
<td>X77</td>
<td>Y77</td>
<td>X78</td>
<td>Y78</td>
<td>X79</td>
<td>Y79</td>
<td>X80</td>
</tr>
<tr>
<td>Y80</td>
<td>X81</td>
<td>Y81</td>
<td>X82</td>
<td>Y82</td>
<td>X83</td>
<td>Y83</td>
<td>X84</td>
<td>Y84</td>
</tr>
<tr>
<td>X85</td>
<td>Y85</td>
<td>X86</td>
<td>Y86</td>
<td>X87</td>
<td>Y87</td>
<td>X88</td>
<td>Y88</td>
<td>X89</td>
</tr>
<tr>
<td>Y89</td>
<td>X90</td>
<td>Y90</td>
<td>X91</td>
<td>Y91</td>
<td>X92</td>
<td>Y92</td>
<td>X93</td>
<td>Y93</td>
</tr>
<tr>
<td>X94</td>
<td>Y94</td>
<td>X95</td>
<td>Y95</td>
<td>X96</td>
<td>Y96</td>
<td>X97</td>
<td>Y97</td>
<td>X98</td>
</tr>
</tbody>
</table>
References


