GEOGRAPHIC REGIONS:
A NEW COMPOSITE GIS FEATURE TYPE

Jan van Roessel and David Pullar
ESRI
380 New York Street
Redlands CA 92373
jvanroessel@esri.com
dpullar@esri.com

ABSTRACT

ARC/INFO 7.0 will have a new capability to handle overlapping and disjoint areas through a new feature class: the "region." A region consists of one or more non-overlapping areal components. Regions may overlap. Two relationships are maintained between the composite region feature and the base polygonal and chain data: (1) a region-polygon cross reference, and (2) a region-boundary chain list. Regions can be interactively created and edited, or may be constructed in bulk from arc loops. Regions are maintained by other GIS functions such as overlay. Coverages resulting from overlay inherit the region subclasses from the parent coverages. Spatial inquiries can be made against multiple region subclasses within the same coverage through "regionquery" and "regionselect" functions.

INTRODUCTION

Many GIS users are concerned with the management of overlapping and disjoint areas of interest in a single coverage of a vector-based GIS. Often because of the frequency of overlap, and the irregular way in which it occurs, it has not been practical to manage overlap by using different coverages.

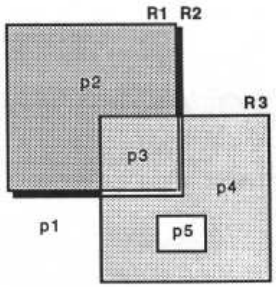
Application areas that can benefit from managing irregular overlapping data in a single coverage are varied and many. Oil and gas applications must keep track of overlapping lease data. They are also concerned with overlapping geological data at various depth levels. Forestry applications must manage stand treatment data, fire histories, and other historical data. Nature conservation deals with natural communities, plant and species habitats, which are not spatially mutually exclusive. Cadastral and parcel mapping must keep track of overlapping historical parcels and legal land records. AM/FM applications may want to manage different floor plans in the same coverage. Applications that have nested data at different levels, such as Bureau of the Census data, are also a prime candidate for an implementation using regions.

The development of the region feature class results primarily from the need to manage overlapping data, but the term region is more associated with areas that may be spatially discontiguous. The new feature class also provides the capability to manage noncontiguous areas with identical attributes as a single region.

BACKGROUND

In the past many ARC/INFO users have implemented some type of scheme to deal with overlapping data by implementing systems using the Arc Macro Language (Gross 1991). The most sophisticated system of this type has been a cadastral-parcel mapping system developed by ESRI-UK. Invariably, all these schemes have used the device of a cross-reference file. This file stores the relation between the overlapping units and the base polygons. Figure 1 show the use of a cross reference file

With the introduction of routes and sections in ARC/INFO 6.0 it became clear that a similar composite type could be constructed based on polygons. Such a feature type would provide "core" support for the overlapping polygon problem, freeing users from having to implement custom made schemes.



Region	Polygon
1	2
	3
2	2
	3
3	3
	4

Figure 1. Cross-reference file example

The basic design question was whether to implement an approach where geometry is shared, or whether to create overlapping polygons with duplicated geometry. Both options are actually available for lines in ARC/INFO. The non-planarity of lines is achieved in two ways: (1) logically through the route and section system, by which multiple coincident routes may coexist with shared geometry, and (2) with duplicated geometry through the existence of coincident arcs and crossing arcs without nodes at the intersection.

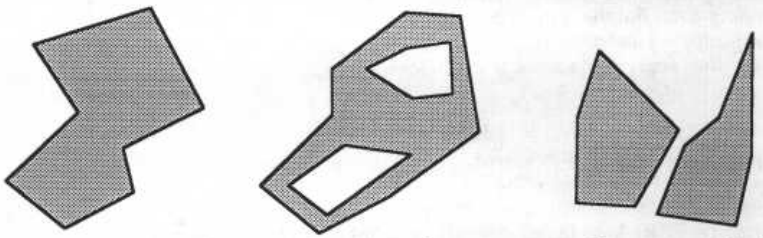
The same options were available for polygons. We selected the shared geometry approach, with multiple subclasses. This is equivalent to the route and section system. One important reason for choosing the shared geometry approach is cadastral applications, where the basic geometry must not change when parcels are subdivided.

LITERATURE REVIEW

It is apparent that the term region has many definitions in geography. Therefore, it is inappropriate for us to give a formal or encompassing definition for this term. Rather, we concentrate on where region is defined as a spatial unit in a GIS.

The spatial units used conventionally to represent discrete geographic entities are points, lines, and polygons. Spatial units are classified based upon their dimensionality and geometric properties. Laurini and Thompson (1992) define spatial units for polygons and regions. The definitions, see figure 2, are as follows;

- a simple polygon is a connected 2-dimensional space bounded by lines,
- complex polygon is a connected 2-dimensional space bounded by lines containing one or more holes,
- region is made up of two or more separate simple polygons that may be disjoint.



Simple polygon (no holes or exclaves) [cell]

Complex polygon (has exclaves) [area]

Non-connected polygon (has exclaves) [region]

Figure 2: Polygons and regions from Laurini and Thompson

Laurini and Thompson also describe spatial units from the perspective of how they are combined. In particular they define a compound type as an object created by combining spatial units of the same type. A combination of polygon objects forms a new compound type that has different semantic properties than the single unit. For instance, a compound spatial type for land parcel may be composed of a polygon for the parcel boundaries and a polygon for a house located on the parcel.

Their definition for region is reasonably consistent with the region feature type in ARC/INFO. The region is used to represent areal entities in 2-dimensional space as a non-connected polygon, but an ARC/INFO region is allowed to have exclaves. Regions are also consistent with the definition for a compound object. A region is a composite made up from the underlying polygons, but it has its own semantic properties. A region has descriptive properties that are independent of the properties for its component polygons.

A more formal definition for regions is given in Scholl and Voisard (1989). A region type has both spatial domain properties and set-theoretic properties. That is, a region type can be considered as the spatial domain for an object, and it also signifies that a region type is a composite made up from a set of elementary subsets of space (which are also regions). Here an elementary region is defined as a subset of R^2. From figure 3 we see it can be any of the following:

- a polygon (e.g. r3, r4)
- a subset of R^2 bounded by lines (r1, r2)
- the union over R^2 of several connected or non-connected parts (e.g., r3 $\cup$ r4)

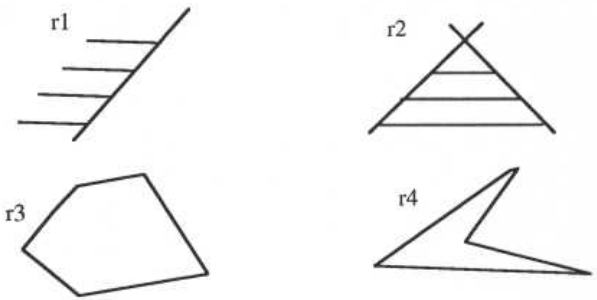


Figure 3. Regions as defined by Scholl and Voisard

This definition of regions by Scholl and Voisard is too general. The regions in ARC/INFO use a stricter interpretation. The subsets of R^2 that are part of a region must be 2-dimensional. Formally we say that the subsets of points have a 2-dimensional neighborhood. This interpretation recognizes the fundamental difference between objects that have a 1-dimensional neighborhood, i.e., linear objects, and those that have a 2-dimensional neighborhood, i.e., areal objects.

ARC/INFO Data Model

The ARC/INFO data model is composed of a number of geographic data sets: coverages, grids, attribute tables, tins, lattices, and images. The coverage data set encompasses several feature classes. Depending upon the user's method of abstracting real world entities, geographic objects are represented as points, lines, or areas. These object types can be modelled in an elementary way as nodes, arcs, and polygons (Morehouse 1992). Elementary types are related in a topological sense to form a partition of a planar surface (similar to the USGS Digital Line Graph). One coverage is then analogous to a surface layer. Combinations of elementary geometrical types are made to build more elaborate representations of geographic objects. These are called composite feature classes. They include sections, routes and regions. The sections and routes feature classes are used to

build route systems on top of arcs and nodes to define dynamic models over a network. Regions are a new feature class that provides a similar capability as routes, but build arbitrary areas on top of elementary polygons.

Figure 4 is a schematic of the relationships between feature classes in a coverage data set. The figure shows that composite feature classes are built upon elementary feature classes. This allows users to build integrated coverages as combinations of the elementary types. Structurally a region is composed of many polygons and the set of arcs that form the exterior boundary of the region. The arcs are related to nodes by the encoded connectivity relations (to/from), they are related the polygons by the encoded coincidence relationships (left/right), and may also be related to a route system encoded as composite relationship.

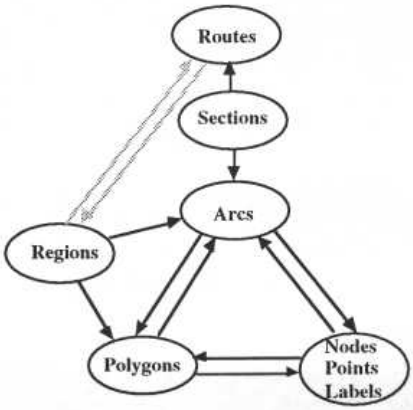


Figure 4. Regions and ARC/INFO Data Model

The relationship between polygons and regions can be understood by the ownership ordering for areal spatial units. A hierarchical tree graph is used to illustrate these relationships (Laurini and Thompson 1992). Figure 5 shows how regions and polygons can be modeled as a two tier graph.

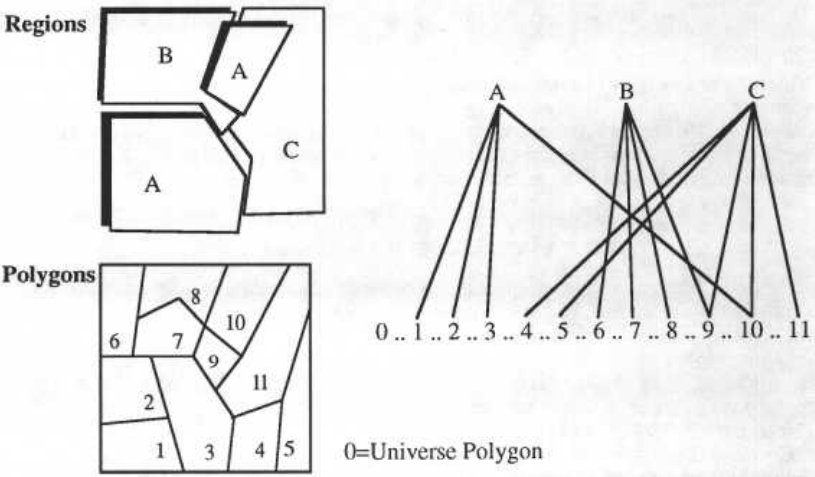


Figure 5. Regions and polygons modeled as a two tier graph

The ordering is a many-to-many and there is no nesting. The lowest level has the elementary spatial units, namely polygons. These units are aggregated to form a composite spatial unit, namely a region, as arbitrary combinations of polygons. Note that polygons partition the plane into a universe polygon and then 1-N elementary polygons. Regions are composed of subsets of the N polygons, but the universe polygon cannot belong to a region. Each region borders on the universe for both internal and exterior boundaries, and a hole within a region belongs to the universe. This allows boolean logic operations on regions that cannot be expressed on a polygonal partition in ARC/INFO where each connected component is always covered by non-universe polygons.

Another important aspect of the ARC/INFO data model is that each feature class is associated with an attribute table. For elementary feature classes there are separate feature attribute tables for nodes, arcs and polygons in a coverage data set. Composite feature classes allow several attribute tables to be associated with sections, routes, or regions. For this reason the individual attribute tables for composite classes are referred to as subclass tables. The way a user organizes a coverage is to have one composite feature subclass for each homogeneous set of attributes. In other words, a set of regions with common attributes is assigned to the same region subclass. For example, within an integrated coverage one region feature subclass may be used to represent types of forest with cover attribute information, and another region feature subclass may be used to represent flood hazard areas with flood level and date attribute information. For each region instance in a composite feature there is one record in the subclass table. Figure 6 shows a schematic of how regions are related to attribute tables.

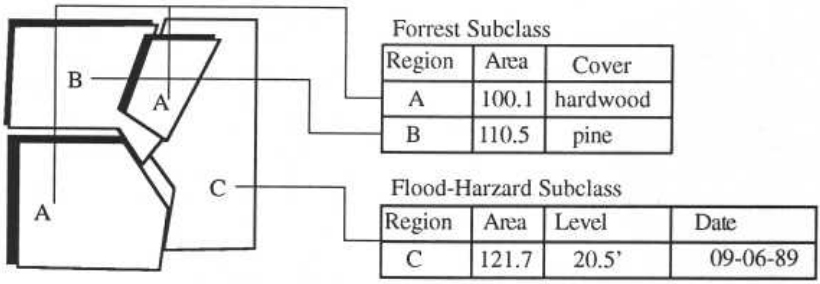


Figure 6. Region subclasses and attribute tables

The storage structure for regions uses three files, a PAL file to store the region-arc relations, a RXP file to store the region-polygon relations, and a PAT file to store region attribute information. The first four fields of the PAT file are maintained by the system, they include the record number, user identifier, area, and perimeter. Since there may be many region subclasses within a coverage a naming convention is adopted. The PAT file is uniquely identified as **<cover>.PAT<subclass>**, the PAL and RXP files follow this convention.

FUNCTIONAL MODEL

The functionality of the ARC/INFO system resides in a number of functions operating on coverages. Different actions occur based on the feature types present in the coverage. With spatial features such as regions and polygons new spatial features may result. Newly generated polygons must always be assigned to a new coverage, because polygons are mutually exclusive. The unique overlap quality of regions does not pose this constraint, so that regions may stack up in the same coverage.

ARC/INFO's GRID system similarly has the concept of "stacked" grid data, but one of the unique differences between rasters and regions is that the depth of the stack is uniform for a grid, but it may be variable for regions.

The functional mode for regions takes advantage of the fact that regions may accumulate in a coverage.

Let $Cov(A)$ denote a coverage with a set of region subclasses A. F_b is a binary operator that produces one output coverage from two input coverages. The region functionality for this type of function is:

$$Cov3(C) = Cov1(A)\ F_b\ Cov2(B)$$

where $C = A \cup B$. If A and B contain an identical subclass s, each with region sets P_S and Q_S, then C will have subclass s with a region set $R_S = P_S \cup Q_S$. With identical subclass, we mean that the subclass bears the same name and has the same attributes, and therefore has the same schema. The function will fail for subclasses with identical names and different attributes. Examples of F_b are the ARC/INFO functions UNION, INTERSECT and IDENTITY.

A second type of binary function is G_b, with the model:

$$Cov3(A) = \mathrm{Cov1}(A)\ G_b\ \mathrm{Cov2}(B)$$

where the region subclasses from the second cover do not enter into the output cover. Examples of this type of function are CLIP and ERASE.

Another class of functions is the unary function $F_u(A)$ operating with single coverage input and output. Here the subclass set A has a single subclass s. This function produces regions in subclass s, and has the overall effect:

$$Cov2(C) = F_u(A)\ Cov1(B)$$

where $C = A \cup B$. Examples of this type of function are REGIONQUERY REGIONBUFFER. For instance, in the case of REGIONBUFFER, B may be an empty set and buffer regions are produced from points, arcs or polygons into output subclass s.

For functions of type F_u, A and B may have a subclass with an identical name, but different attributes. The function will not fail, as with F_b. Denoting a subclass schema of subclass s by $\{name, I\}$ where I is a set of attributes, then if A contains a subclass s with $\{sname, I\}$ and B contains a subclass q with $\{sname, J\}$ then C will have a subclass r with $\{sname, K\}$, where $K = I \cup J$.

For a function of type F_b, attributes rows of regions in subclasses with the same schema are simply appended. For a function of type F_u, attribute rows are appended, inserting blanks or zeros, as appropriate, into the values for the attributes I.

CONSTRUCTING REGIONS IN BULK

Regions can be constructed in a number of ways. They can be created interactively, or they can result from operations on other regions, or may be created in bulk, starting with arc data.

A special function named REGIONCLASS builds "preliminary regions" out of arc data. The preliminary regions produced by the REGIONCLASS function groups a set of arcs based on a user specified attribute. Each group of arcs must form one or more rings for one region. The rings are constructed and recorded in a region PAL file.

The preliminary regions are then converted into fully built regions in the CLEAN process. The arcs are first intersected and then enter a planesweep polygon construction phase in which polygons are built from intersected line segments. Each line segment has a set of associated regions that is propagated in the planesweep process. This produces the region cross reference file for each subclass. The cross reference files are then used in turn to update the region PAL files for each subclass. This completes the process of building topologically structured regions.

EDITING REGIONS

Regions with the ARC/INFO ARCEDIT system can be edited as a collection of arcs or as a collection of polygons. ADD and DELETE commands add to the current region, or

delete from the current region the primitive feature's type selected. If arcs are edited the selection commands require the user to select the region(s) by pointing to the arc(s) that belong to that region. Similarly, if editing polygons, the selection commands require the user to select the region(s) by pointing to the polygon(s) that belong to that region.

When the coverage is saved, each region subclass is updated to reflect the changes. In either case, the PAL file is updated to reflect the new arc and node numbering scheme. If polygon topology is maintained, then the RXP file is update to reflect the new polygon numbering.

REGIONS IN OVERLAYS

Coverages that are the result of overlays inherit the region subclasses of the input coverages. For the UNION, INTERSECTION and IDENTITY functions, the output coverage receives the subclasses from both input coverages. For the ERASE and CLIP functions, region subclasses from the erase or clip coverages are not inherited. For an ERASE, the area of the ERASE cover is blank, and for a CLIP, only the outline of the CLIP cover is used for clipping.

The extent of regions appearing in the output coverages is limited by the extent of the overlay result. This is shown in Figure 7. As a result of the overlay operation, subclasses may become empty, but they do not vanish.

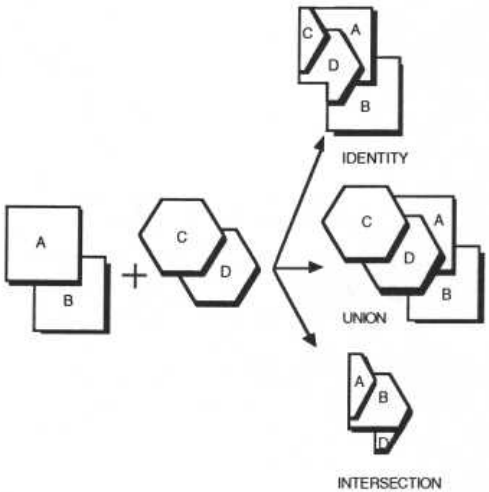


Figure 7. The classical overlay operators and their effect on regions.

The region subclass inheritance mechanism for overlays revolves around a fast updating of the subclass RXP files, using the polygon parentage files produced by the overlays. If an input polygon P has an associated region set R, and another input polygon Q has an associated region set S, and P intersects Q, then the region set for the intersected polygon is the union of the region sets of the parent polygons. This is illustrated in Figure 8. The updated RXP files are then converted into updated region PAL files.

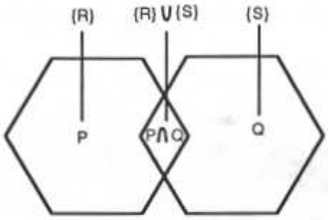


Figure 8. Region set associated with intersected polygons

One way in which regions may be used is to do "integrated coverage" analysis. Instead of keeping the various polygonal thematic layers in separate coverages, they can be integrated into a single coverage as region subclasses. Analysis and queries are then performed with much greater speed in the integrated coverage, provided of course that the base data do not change. Some precision may also be lost due to fuzzy tolerance processing for the integration. An easy way to integrate two polygon coverages is to copy the polygons of each input coverage into regions using the POLYREGION command. This is then followed by a UNION overlay. The resulting integrated cover will show the input coverages as region subclasses.

REGIONS AND "DISSOLVE"

The counterpart of intersecting polygons in overlays is merging of polygons in functions such as DISSOLVE. In that case, if we have a polygon P with region set R and a polygon Q with region set S, and P is combined with Q into a single polygon, the region set for P $\cup$ Q becomes R $\cap$ S.

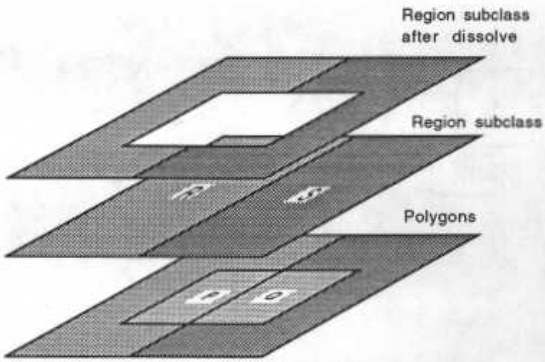


Figure 9. Effect of merging polygons on regions

An example is shown in Figure 9, where polygons P and Q have single associated regions R and S. Merging polygons P and Q yields an empty region intersection set, so that the region subclass after the dissolve develops a "hole."

The above example shows how traditional polygon dissolving has region implications. It is also possible to merge and dissolve regions. Dissolving polygons merges adjacent polygons that have the same value for a specified item. Because regions can be discontiguous, adjacency plays no role when dissolving regions. The following example shows how regions are dissolved. Assume a region subclass with regions shown in Figure 10a.

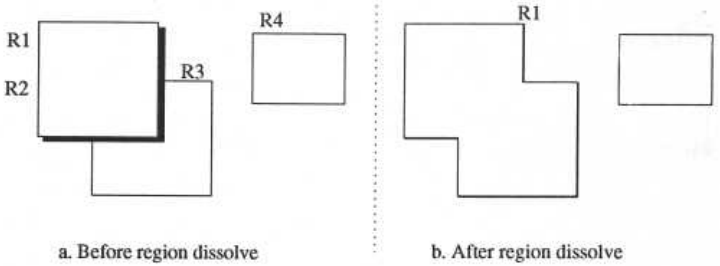


Figure 10. Region dissolve example

Regions R1 and R2 are two identical overlapping squares. Assume further that the subclass feature attribute table has an attribute that has the same value for each region. Using REGIONDISSOLVE with this attribute creates one output region as shown in Figure 10b. Note how the dissolve not only occurs in the horizontal dimension, but also takes place vertically by removing overlap.

ANALYSIS WITH REGIONS

While the initial impetus for implementing regions was to keep track of, and manage overlapping areal units, it became clear that regions may be used for unique types of analysis. At the moment we have implemented two functions, REGIONQUERY and REGIONSELECT, but there are other opportunities that may be exploited in the future.

REGIONQUERY is an ARC function that can mimic the ARC/INFO "classical" overlay functions in an integrated coverage. But this is just one of its capabilities. While the "classical overlay functions" are two at a time functions, REGIONQUERY can perform boolean selection against any number of region subclasses. In addition to selection, it also produces output regions that are homogeneous with respect to a number of user specified output items.

REGIONQUERY usage can be represented as:

<out_subclass> {WITH <in_subclass.attribute...in_subclass.attribute>}

WHERE <logical_ expression>

where *<out_subclass>* is the output subclass, pre-existing or to be created, *<in_subclass.attribute>* is an attribute of *<in_subclass>* that will be added to the schema of *<out_subclass>*. The set of these attributes forms the output attribute list. The list is optional.

The *<logical_expression>* is a completely general boolean expression where the operands are either constants or attributes of the form *<in_subclass.attribute>*. A special pseudo item *$subclass*, meaning "where the subclass exists" (no-zero record number) can also be used. The logical expression used defines the "territory" qualifying for the query, while the output attribute list defines the "granularity" of the output. Territory in the region's case also extends to the vertical dimension, and the same is true for granularity, because identical overlapping regions with identical output attribute values are collapsed to a single region (see the previous "dissolve" discussion).

The REGIONQUERY function also has an option to produce contiguous output regions.

The model under which REGIONQUERY operates is the following. Each polygon in the input coverages has a number of regions for a given subclass of which it is a part. These are the regions "above" the polygon. The polygon and the regions above it form a subclass "stack."

If only a single subclass is involved in the query, the selection expressions are evaluated for each member of the stack, and if true, the polygon is selected to be a part of a candidate output region corresponding to that member of the stack for which the expression is true.

If multiple subclasses are present, and a single polygon has multiple subclass stacks with more than one region, REGIONQUERY will combine the stacks in the form of a cartesian product, and evaluate the attributes according to the logical expressions for each member of the product.

Output items are assigned to each selected cartesian product combination and a dissolve is performed to make the attribute value combinations unique for each output region.

Example 1

This is a traditional site selection example. The objective is to select a proposed laboratory site where the preferred land use is brush land, the soil type should be suitable for development, the site should be within 300 meters of existing sewer lines, beyond 20 meters of existing streams and be at least 2000 square meters in area.

Assuming an integrated coverage with a land use, soils, stream buffer and sewer buffer subclasses we can pose the following query using the contiguous region option:

potential_sites WHERE landuse.lu-code eq 300 and soils.suit >= 2 and not $streambuf and $sewerbuf

This is followed by a second query using the *potential_sites* subclass:

qualifying_sites WHERE potential_sites.area > 2000.0

Example 2

A typical problem concerning regions or polygons is to find out which units of one class are completely or partially contained in another class. Containment is related to overlap, where 100% overlap means fully contained in (or a duplicate of) and 0 % overlap means the opposite. In general one can ask the question display all units of class X that are p% contained in class Y, where 0 <= p <= 100.

Assuming that we have an integrated coverage with "floodplain" and "vineyards" region subclasses we can solve the problem of selecting all vineyards that are at least 80% contained within the floodplain with two queries:

overlap WITH vineyard.vineyard# WHERE $floodplain and $vineyard

This creates regions that are the intersection of the floodplain and vineyard subclasses in a new subclass called overlap. They will have as an attribute the record number of the vineyard region (vineyard.vineyard#).

Then we do the following query:

mostly_within WHERE (vineyard.vinyard# = overlap.vineyard#) and
(overlap.area / vineyard.area >= 0.80)

Here we use the unique region overlap property to compare the areas of two regions that share a common overlap area. The output regions in the "mostly_within" subclass are those vineyard portions that overlap the floodplain and are at least 80% of the original vineyard size.

Example 3

Given that we have a set of overlapping lease data stored in a region subclass called "leases" that has an attribute "leascholder" we can generate a report of overlap pairs by first making a copy of the lease region subclass, and name it "leasecopy." Then we run the following query:

conflict WITH lease.leaseholder leasecopy.leaseholder
WHERE lease.leaseholder <> leasecopy.leaseholder

When REGIONQUERY is confronted with one or more subclass region stacks over a polygon, it makes the cartesian product and guarantees that the output will have a region for each unique overlap combination where the leaseholders are different. The attribute

table of the output subclass "conflict" will contain a unique combination of two different leaseholders that have overlapping leases in each row.

REGION DISPLAY AND SELECTION

The ARC/INFO ARCPLOT REGIONSELECT function is a companion function to REGIONQUERY. Unlike REGIONQUERY, it does not create new output regions. The user specifies output subclasses to which the selection refers, rather than output attributes. Like REGIONQUERY, the user also specifies a logical selection over multiple region subclasses. Regions in the specified set of subclasses that have attribute values in their overlap area for which the logical expression is true are selected. As in REGIONQUERY, polygons may be used as another region subclass.

One use of REGIONSELECT is to identify all regions of which a selected polygon is a part. REGIONSELECT operates in the ARCPLOT selection environment, and therefore interacts with past selections.

REGIONSELECT may be also be used in place of REGIONQUERY, if the output regions are identical to already existing regions. For instance, in the second example above, REGIONSELECT may be substituted for the second query.

Regions can be displayed similarly to polygons. A REGIONLINES command can be used to display regions with offset lines, so that overlapping regions that are identical in shape and size can be differentiated.

There may also be other as yet unexplored display techniques that would more effectively communicate impressions of overlapping areal data to a user. A model might be used in which region shades have gradients across a region, so that a region's identity is uniquely shown in a transparent display model.

CONCLUSION

We believe that the ARC/INFO 7.0 release will have well rounded initial capability to handle disjoint and overlapping data. However, there may be requirements and uses that will only become clear when the new feature class is applied in practice.

It is also interesting to speculate how regions may fit in with future GIS development. A potential application may be in the area of fuzzy GIS data (Heuvelink and Burrough, 1993). The traditional polygonal model represents a prismatic probability density function with a 100% certainty that the attribute occurs within the prism contour. With a fuzzy data model overlapping regions may be used to store various levels of a probability density function pertaining to the probability of the presence of an attribute. A corresponding region modeling capability may be needed to manipulate attributes and their associated probabilities.

Another used for the feature class may be in the support of partially ordered sets and lattices (Kainz et al, 1993). Partially ordered sets defined on regions might be used to handle a larger variety of containment queries.

REFERENCES

Gross T. 1991, "Modeling Polygons Located at Several Positions on a Z-Axis as a Single Planar Map." In Proceedings, Eleventh Annual ESRI User Conference, Vol. 2, pp. 5-20.

Heuvelink G.M.B., and Burrough P.A. 1993, "Error propagation in Cartographic Modelling Using Boolean Logic and Continuous Classification." International Journal of Geographic Information Systems, Vol. 7, pp. 231-246. Taylor & Francis, London, Washington D.C.

Kainz W., Egenhofer M.J., and Greasley I. 1993, "Modelling Spatial Relations with Partially Ordered Sets." International Journal of Geographic Information Systems, Vol. 7, pp. 215-229. Taylor & Francis, London, Washington D.C.

Laurini R., and Thompson D. 1992, "Fundamentals of Spatial Information Systems." Academic Press, London.

Morehouse S., "The ARC/INFO Geographic Information System." Computers and Geosciences, Vol. 18, No. 4, pp. 435-441

Scholl M., and Voisard A. 1989, "Thematic Map Modeling." In Proceedings, Symposium on Very Large Spatial Data Bases. L.N.R.I.A., 78153 Le Chesnay, France.