Chapter 10. Map Overlay and Geoprocessing

Objectives

- Learning about spatial analysis functions, including overlay, clipping, and buffering
- Using map overlay to analyze multiple spatial criteria
- Understanding differences between spatial joins and overlays
- Geoprocessing with menus, ArcToolbox, and Model Builder

Mastering the Concepts

GIS Concepts

Over the years many procedures have accumulated for characterizing spatial relationships within and between spatial features. Functions can find areas shared by two or more conditions, evaluate distances, extract or erase areas of interest, merge similar features together, examine distance relationships between points, and more. The entire enterprise of GIS, including gathering the data, putting them in digital form, designing the geodatabase, managing the data, and creating maps, is justified by the ability to apply these tools to extract information that might be difficult to obtain any other way. GIS shares many capabilities with CAD systems and database software, but spatial analysis gives GIS unique power to get the most out of map and attribute data.

Chapter 9 discussed spatial joins as one way to analyze spatial relationships. In this chapter, we'll learn more tools to solve spatial problems. Often two or more functions will be strung together to solve a specific problem, a practice called **geoprocessing**. We'll begin with a general discussion of spatial functions and then investigate how these functions are implemented within ArcGIS.

Map overlay

Map overlay combines two layers to create a new output feature class containing information from both of the inputs. In Figure 10.1, the polygons from a geology layer and from a land ownership layer have been combined to produce a new set of polygons. Each new polygon has

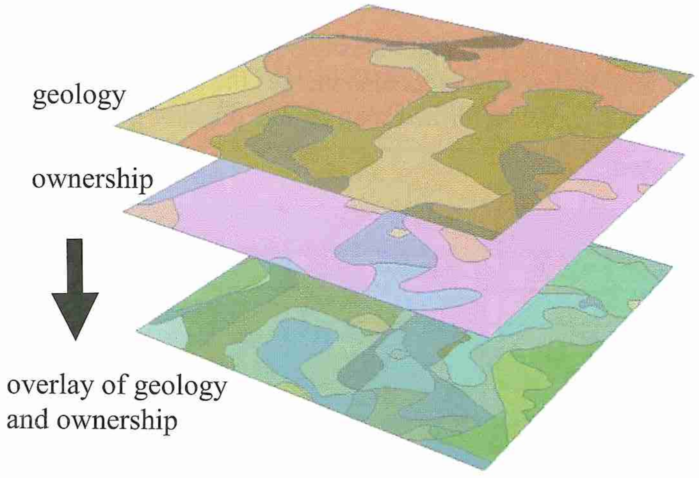

Fig. 10.1. Map overlay combines two feature classes to create a new feature class containing information from both inputs. Both features and attributes may be combined. Here, each new polygon in the output contains both geology and ownership attributes.

been given the attributes of the originals, and each contains information about both geology and ownership for the same feature. The output layer might be useful for selecting areas containing certain gem-bearing rock formations on public land that would be suitable for mineral hunting.

Overlay functions fall into two categories, those that do not combine attributes and those that do. Functions that do not combine attributes, also called extraction functions, include **clip** and **erase,** which were discussed in Chapter 8. In extraction, the input features retain their attributes, but the attributes of the clip or erase boundary layer are ignored. The two functions that combine attributes include **intersect** and **union**. Union preserves all of the features from both inputs; intersect only keeps features common to both inputs.

Overlay with attributes

The intersect and union functions are related to spatial joins in that they correlate features based on their spatial relationships to each other. However, spatial joins fail when spatial features do not overlap exactly (recall the creeks and geology example from Chapter 9). Consider the map of roads and land use shown in Figure 10.2. The state government has requested a report of the total miles of road falling into each land use category. At first glance, a spatial join might be considered to generate this information; by joining the land use polygons to the roads, one might expect to get a field showing the land use type each road crosses.

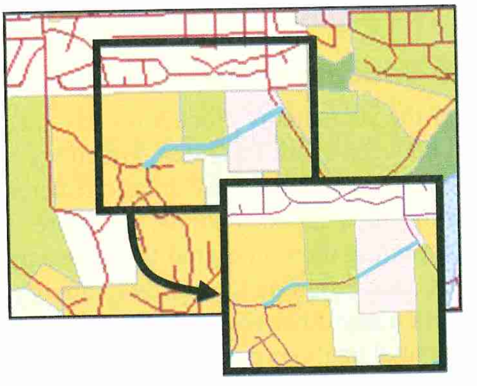

Fig. 10.2. Splitting a road to assign attributes to each piece

However, the single selected road, highlighted in Figure 10.2, crosses three different land use classes. How would a land use type be assigned in this case? If one used the *completely within* join option, no land use class would be assigned for the road because it does not lie entirely inside a polygon. A *summarized join* would do no good because a nominal data type such as land use class can't be averaged. The ideal solution would split the road into three sections and assign the land use to each section, and this is what overlay functions do, thereby enforcing a one-to-one relationship between features and enabling a perfect correspondence when joining the tables. In Figure 10.2, the original single road is now three sections, and each segment in the output table (Fig. 10.3) is assigned the attributes of the land use polygon containing it.

Map overlay with attributes occurs in two forms. **Union** combines two polygon layers, keeping all areas and merging the attributes for both layers. **Intersect** also merges the attributes but retains only the areas common to both layers, and may be performed with points, lines, or polygons.

Fig. 10.3. During overlay, each new road segment retains its original attributes and receives the land use code and other attributes of the polygon in which it falls.

roadlu

OBJ_ID	ROADNAME	SUFFIX	LU-ID	LU_CODE
3033	2 ST	ST	153	High Density Residential
3034	QUINCY ST	ST	153	High Density Residential
3035	RANGE RD	RD	72	Office/Commercial
3036	RANGE RD	RD	305	Public
3037	RANGE RD	RD	70	Medium Density Residential
3038	SOO SAN DR	DR	72	Office/Commercial

A **union** creates all possible polygons from the combination of features in two polygon layers. Figure 10.4 shows a union performed on geology and slope class layers to create a landslide hazard map. Landslide risk depends primarily on two factors, the strength of the geological units (sandstones are less susceptible to sliding than shales) and the slope. The lighter areas in the slope class map have lower slopes, and the dark areas have higher slopes. A union produces a new feature class with new polygons, each of which possesses the original attributes of its parents. The new feature class can then be evaluated for the various combinations of slope and geology that may constitute a landslide hazard.

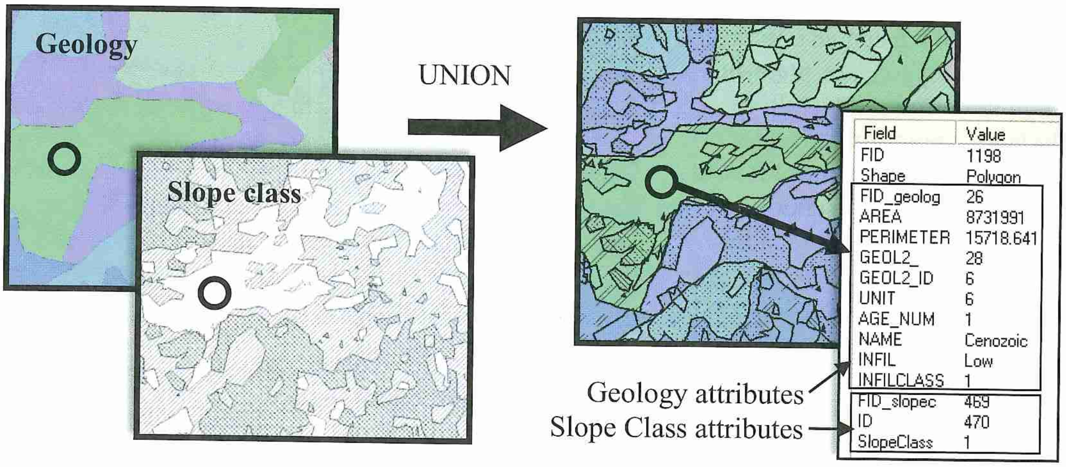

Fig. 10.4. A union creates every possible new polygon from the combined input layers. Each new polygon assumes the attributes from both layers; the new polygon marked with the circle contains attributes from both the geology and slope class layers.

The **intersect** operation with two polygon layers is similar to a union, except that it only keeps the polygon areas that are shared by both input layers. This function provides a way to find out where two or more conditions hold simultaneously, the foundation of a class of problems known as **suitability analysis**—evaluating a landscape to find which areas best serve a given purpose based on a set of factors. For example, one could find the potential habitat for a species with specific environmental requirements.

Imagine that a rare species of snail inhabits the Black Hills. This snail prefers limey soils in cool and dense coniferous forest and is rarely found above an elevation of 1600 meters or below 1200 meters. Each data set (elevation, geology, vegetation) is queried to find the areas meeting the specified condition, and the resulting three layers are intersected to find where all three conditions hold (Fig. 10.5). The habitat polygons in the output contain the full attributes of all three layers,

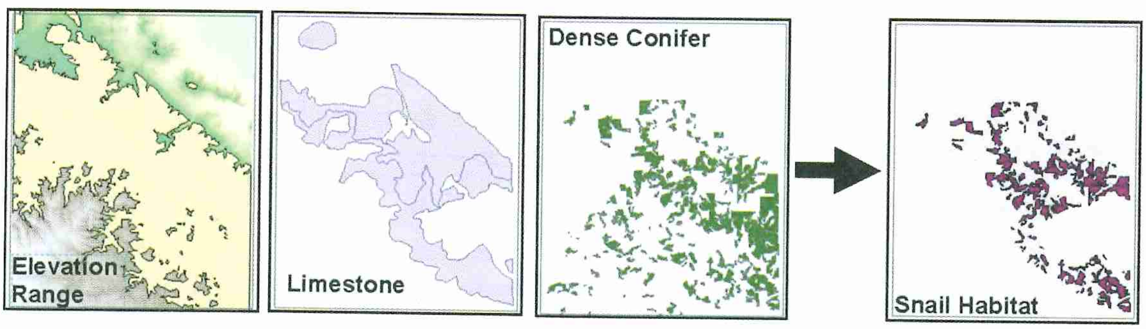

Fig. 10.5. Intersecting elevation, limestone areas, and dense conifer vegetation can help identify areas of potential snail habitat.

should that information be needed. Such a map would help biologists create sampling strategies for counting populations, analyze whether the habitats are interconnected or widely separated, and make decisions about forest management to protect the snails.

The snail habitat demonstrates polygon-on-polygon intersection. Intersection can assign attributes from polygons to lines or points within them. The roads and land use problem described in Figure 10.2 is an example of line-in-polygon intersection, assigning land use categories to the roads. In Figure 10.6a, line-in-polygon intersection is used to assign a geologic unit to each stream reach (as in Chapter 9), useful for estimating stream loss to groundwater. One can also intersect points with polygons, or point-in-polygon intersection. A realtor might have a point feature class of houses for sale and intersect it with polygons representing school districts in order to be able to list the school for each house (Fig. 10.6b). Note that a point-in-polygon intersection is equivalent to a simple inside join with the point layer as the destination, even though they use different tools to do it.

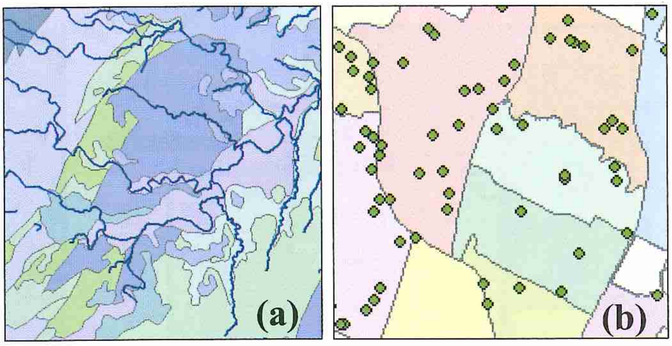

Fig 10.6. (a) Assign geologic units to streams with line-in-polygon intersection; (b) assign schools to houses for sale with point-in-polygon intersection.

Whereas the union function requires two polygon input layers, the intersect command is more versatile. Both input and overlay layers may contain points, lines, or polygons. The output geometry may vary, but it cannot exceed the dimensionality of the lowest input.

If the inputs are both polygons, the output may be points, lines, or polygons. In Figure 10.7a, the two circles might represent the drug-free zones around two schools. The polygon output shows the area of overlap; the line output could become a boundary on a map.

If the inputs are both line feature classes, the output may be lines or points (Fig. 10.7b). These lines might represent hiking trails, in which case the line output would find locations where two trails travel together. The point output could find trail intersections that need to have signposts placed on them.

If the inputs are polygons and lines, the output may be points or lines (Fig. 10.7c). If the circle represents the area around a school where sex offenders are forbidden to live, the line output could provide a list of the streets and address ranges that are

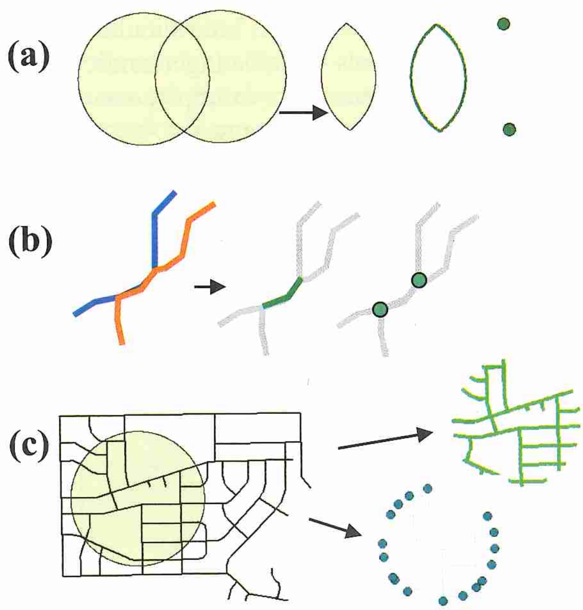

Fig. 10.7. Intersect geometries. (a) Polygons yield polygons, lines, or points. (b) Lines yield lines or points. (c) Polygons and lines yield lines or points.

off limits. If the city intends to put a sign on every street entering the zone, the points output would estimate the number of signs needed and where they should go.

Intersecting two sets of point data requires the output to be points, but is generally not a common analysis, unless someone is looking for duplicate points in two different data sets.

Comparing overlay functions

Figure 10.8 summarizes the different overlay operations that can be performed using two polygon input layers, including the extraction functions clip and erase, which do not combine attributes, and the intersect and union functions, which do. Clip and erase can also accept point and line feature classes for one input, but the boundary input must contain polygons. The output feature classes will have the same geometry as the input classes.

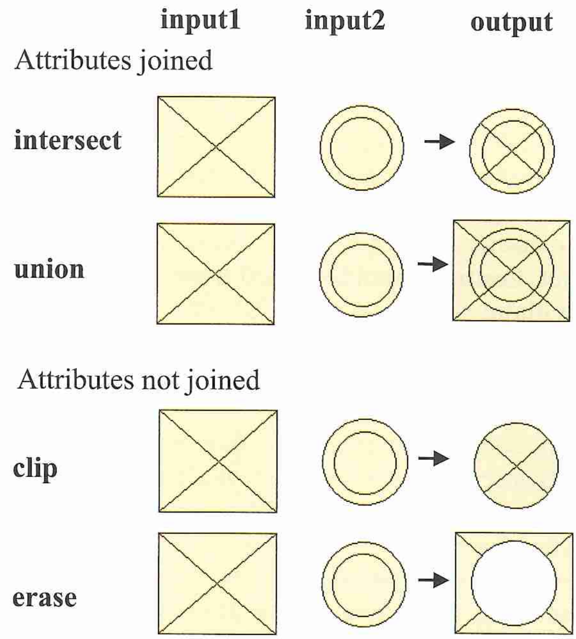

Fig. 10.8. Summary of polygon overlay operations and results

Unions and polygon-on-polygon intersections are similar, so a word about when to use each may be helpful. The primary use of intersect is to find areas where certain conditions overlap. Typically the input layers occupy different regions, as do the elevation range, limestone, and conifer layers shown in Figure 10.5. The goal is to find the areas common to the inputs. Although the attributes from each input are combined in the final table, the attributes themselves are sometimes of little interest. In the snail habitat problem, the pertinent attributes were already preselected using queries. The main goal was developing a map of the overlaps.

In a union, the primary goal is to combine the tables. Generally the input layers all fully occupy the same region, as the geology and slope class layers do in Figure 10.4. It is expected that the entire map area will remain, but that the new features will contain the attributes of both inputs. After a union, the usual next step is to perform attribute queries to find areas with specific combinations of attributes, or to symbolize the combinations to show areas of interest. In the landslide hazard example, we might have extracted the shale and high slope areas and done an intersect to show where the high-hazard areas exist. The hazard is either there or not there. With a union, though, we are able to analyze different combinations of geology and slope to assess potential impacts on slope stability.

Slivers and tolerances

Overlaying layers often produces small extraneous polygons or lines called **slivers** (Fig. 10.9). Sometimes these slivers represent real features. However, more often they arise when overlaying layers share some boundaries, such as voting districts and counties. In theory the shared boundaries should match exactly, but in practice few data sets have been

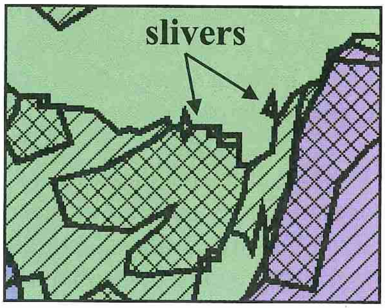

Fig. 10.9. Overlay may produce small sliver polygons.

corrected to this level of integrity. These slivers do not represent real combinations of values and are a nuisance when calculating statistics or performing subsequent operations. It is often desirable to prevent slivers when overlaying.

An **XY tolerance** can be set during an overlay operation. This tolerance specifies the minimum distance between vertices, and it will combine vertices that fall close together. Setting an appropriate tolerance during an overlay will eliminate many slivers. However, the tolerance applies to all vertices, and using a tolerance may degrade the accuracy of the data set. Caution should be used in setting tolerances that are large enough to correct slivers without shifting other vertices an unacceptable amount. It must also be remembered that outputs from an overlay that uses a tolerance generally have a lower geometric accuracy than the original layers.

Slivers are especially troublesome when multiple overlay or analysis operations are planned, because the errors propagate and multiply with each successive step. If XY tolerances are being used, the generalization of the boundaries becomes progressively more severe and may significantly affect the geometric accuracy of the results. It is impossible to avoid these impacts, so users should be aware of what is happening and have realistic expectations for the spatial accuracy of the final output. If an XY tolerance is used during processing, its value must be included in metadata for the final output.

Other spatial analysis functions

Overlay is only one type of spatial analysis available in GIS systems. The remainder of this chapter presents additional commonly used functions, including dissolving, buffering, appending, and merging. All of these analysis functions, and many others, are found in ArcToolbox.

Dissolve

A **dissolve** is used to group features together based on whether they share the same value of an attribute field. For example, the road segments in Figure 10.10 have the same street name (Main St), but they are separate features. A dissolve based on the street name field would yield a new file in which all streets having that name were one feature. A dissolve can also be used to remove lines between polygon features that share the same value. For example, the map in Figure 10.11 shows ponderosa pine stands of different ages. Dissolving on the cover type field removes the age boundaries and yields polygons based on the cover type (ponderosa) only.

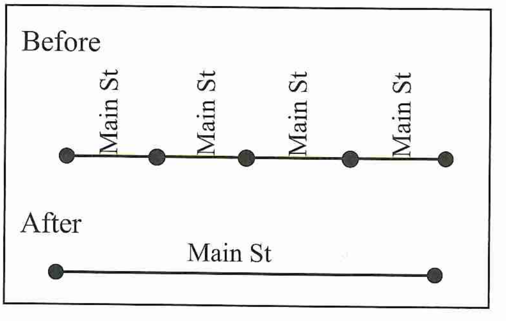

Fig. 10.10. Separate road segments were combined using the dissolve function.

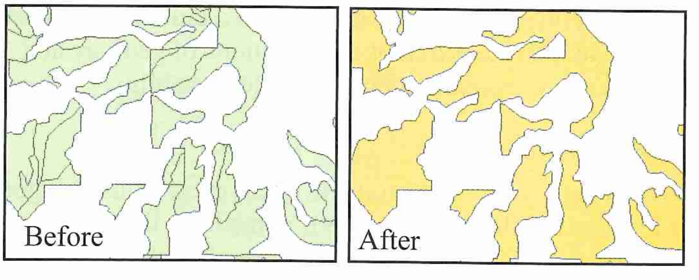

Fig. 10.11. A dissolve removes boundaries between polygons with the same attribute value, in this case, tree species.

The output from a dissolve operation may produce either single features or multifeatures. A multifeature exists when multiple unconnected areas constitute a single feature, as when the seven Hawaiian islands are combined to produce one state feature. In Figure 10.11, the green polygons constitute many individual features. After the dissolve, if the multifeature option was selected, than all of the orange polygons will belong to one feature. If the multifeature option was declined, then the data set will have many orange features.

When features are dissolved, the output layer is a new file containing the single attribute on which the dissolve was based. However, the user can specify additional fields to summarize the information from the original features. For example, in the dissolve shown in Figure 10.11, one might request the average canopy percentage of each output polygon, based on averaging the canopy percentages of the polygons before the dissolve.

Buffer

A **buffer** is constructed to delineate areas that fall within a certain distance of a set of features. Buffers can be created for points, lines, or polygons (Fig. 10.12). They could be used to find 300-yard drug-free zones around schools or sensitive protected areas within 100 meters of a stream. Negative buffers can be applied to determine setback limits from the edge of a piece of property. Buffers can be created as simple rings or as multiple rings. An attribute can even supply different sizes of buffers for different features—for example, buffering primary roads by 200 meters and secondary roads by 100 meters.

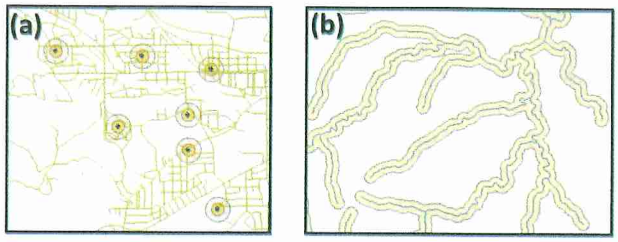

Fig. 10.12. Examples of buffers around (a) wells and (b) roads

Buffers are created for each individual feature and may overlap. This option might be appropriate in analyzing the distribution of soils in buffers around wells. In many cases, however, it is best to dissolve the buffers, which gets rid of the boundaries between them and removes the overlapping areas to create a single region. Figure 10.13 shows the difference in results obtained when the buffers around roads are not dissolved versus when they are. If the areas of the buffers are of interest, then dissolving is always the correct action. If areas were calculated from nondissolved buffers, they would be grossly overestimated, because the overlapping areas would be counted multiple times.

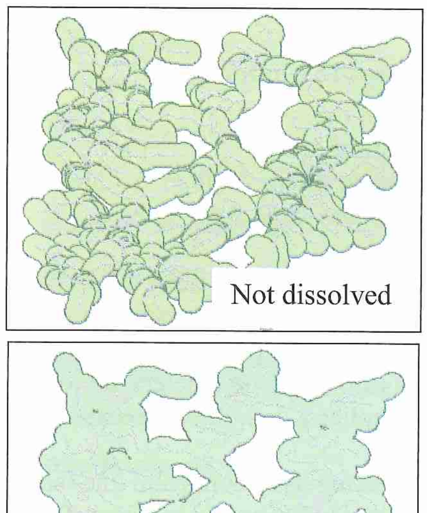

Fig. 10.13. Use the dissolve option to prevent overlapping buffers

Finally, buffering is an intensive process, and the time involved increases quickly as the number of features increases. Any steps to reduce the number of buffers created at one time will facilitate analysis, for example, by performing queries before buffering rather than afterwards.

Append and merge

The **append** function is used to combine the features of two or more layers and place them into an existing target feature class (Fig. 10.14). The appended layers must have the same feature type as the target (i.e., both polygons, both lines, or both points). The two layers must also share the same coordinate system. Overlapping of layers is permitted.

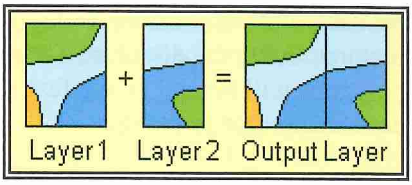

Fig. 10.14. Append combines the features of two adjacent layers.

The treatment of attribute tables during an append requires some consideration. If you wish to combine the attribute information from both layers, the attribute fields must have the same definition and must occur in the same order in both tables. If the two tables differ, one can use the NOTEST option. In this case, if a field in the input layer has the same name and data type as the target layer, then the information will be copied into the target. Fields without matching names will not be carried into the target.

Merge is similar to append, except that it creates a new feature class and offers more flexible treatment of attribute tables. Instead of insisting that the two tables match, it allows the user to specify the fields to be included in the output feature class and which table they will come from.

About ArcGIS

About geoprocessing

GIS analysis involves many functions that operate on data objects, such as feature classes or tables. Geoprocessing applies one or more of these functions in sequence to solve a problem or investigate the properties of data sets. Most ArcGIS tools operate on any allowed data type. In performing a road buffer, for example, one may specify a coverage, a shapefile, or a feature class from a geodatabase. The tools make the necessary allowances to execute the function on the various data types. A few tools operate only on certain data types. An ArcGIS Advanced license, for example, provides the user with a suite of additional tools that work on coverages.

Ways to run tools

Geoprocessing tools may be executed several ways. *Toolbars and menus* provide interactive control of tools. Users may customize menus and toolbars to add frequently used tools that don't appear on the toolbars by default. Users can also create tools and place them on menus, on toolbars, or in ArcToolbox.

ArcToolbox organizes all of the installed tools into one central location. It provides a window with most of the functionality of ArcGIS, containing tools that can be run from ArcMap or from ArcCatalog. Each tool has a set of parameters, or inputs, which must be specified before the tool is run. Some parameters are required, but others are optional, which means that the software supplies a default value that can be changed, if necessary.

ArcMap has a command line that allows the user to type a command and its parameters rather than filling in boxes in a window. Typing requires greater familiarity with the tools, but for experienced users entering commands by typing may be faster and more efficient. Becoming familiar with the command line also paves the way to writing scripts to execute functions. The command line possesses a sophisticated interface to help beginners correctly enter the commands.

Model Builder provides a graphic canvas to string tools together and execute them in sequence. Models have several advantages. First, they record the steps and parameters used to execute an analysis in case questions on methodology arise later. Second, models can be used to explore the consequences of different parameters on the final outcome. Consider a model that calculates fire danger from precipitation, temperatures, vegetation, and structure locations through a series of tools. With a model, one could redo the analysis based on the current day's precipitation and temperature conditions more quickly than if all the steps were executed interactively, and with a lower chance of mistakes. Models can be shared with other users, and they permit a less experienced user to reliably repeat an analysis set up by an expert. Models can incorporate decision structures, conditional statements, and iteration functions if needed. Finally, models can be saved as scripts to provide a starting point for creating geoprocessing programs.

A **script** is a program that may contain conditional statements (if/then), iterative loops, and other control structures that permit sophisticated analysis. Scripts may be written in one of several programming languages, although in each case the geoprocessing commands are identical and only the control statements differ. ESRI has adopted the readily available language Python as the recommended scripting platform. A copy of Python is installed with ArcGIS, and all ESRI scripting examples and supplemental tools are written in Python. However, users may elect to use any COM-compliant language such as Jscript, VBScript, or Perl.

Geoprocessing environments

The operation of tools and commands is impacted by **Environment settings** specified by the user. For example, the default coordinate system setting is *Same as Input,* meaning that the output file has the same coordinate system as the input file. The user could alternatively specify a particular coordinate system, such as UTM Zone 13 NAD 1983. Under this setting, every output feature class would have the UTM coordinate system regardless of its input system.

Figure 10.15 shows some of the Environment settings. The Workspace setting tells ArcGIS where to initially look for data and to save output data sets. It also sets a scratch workspace where temporary files are stored. The Output Coordinates setting specifies the coordinate system given to any outputs. If it is set to UTM Zone 13N, then every output data set will be stored in that coordinate system.

Environment settings are hierarchical. Application settings are set using the main menu bar in ArcMap or in ArcCatalog, and they affect the operation and output of every tool that is run. (They will not cause changes to files that already exist.) The application settings are saved along with a map document, so it is possible for the settings to be operating without the user's awareness. It is not a bad practice to check environment settings if you are getting unexpected results from the tools. When a new map document is opened, all settings are returned to their defaults.

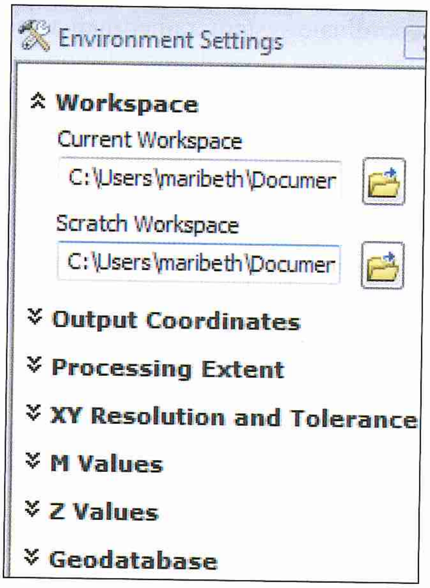

Fig. 10.15. Some Environment settings

All tools in ArcToolbox have an Environments button that can be used to change the Environment settings temporarily for that one run of the tool. For example, the user might have set the application Environment processing extent to a small project area, but needs to buffer some streams and allow the areas to extend beyond the current study area. The Environments button in the Buffer tool can be used to temporarily override the extent setting. Models, scripts, and tools can all have their own environment settings, which take precedence over the application settings.

The default Environment settings work fine for most applications. Users may change them when doing so provides more efficient or accurate work. For example, Amy might have to clip 10 different feature classes and project them from a geographic coordinate system to a projected coordinate system. Instead of running both the Clip and Project tools on each layer, she could set the environment coordinate system to the desired projection and then run the clips, reducing the amount of work.

Coordinate systems

Tools will accept any coordinate system or combination of coordinate systems for input, but rules of precedence dictate the coordinate system of the output. Outputs will be projected on-the-fly, if necessary, according to the following rules:

- If the output is placed in a feature dataset, the coordinate system will always match that of the feature dataset.
- If the coordinate system is set in the Environment settings and the output will not become part of a feature dataset, the Environment settings coordinate system is used.
- If the Environment setting is not set, the default rule applies—that the output will match the coordinate system of the first input to the tool.

When setting the output coordinate system in the Environment settings, you can choose Same as Input (the default), or Same as Display (the current data frame), or set it to match the coordinate system of a particular data set.

Geoprocessing tools are fundamentally spatial in nature and commonly manipulate areas and distances as part of the processing. When geoprocessing, it is best to always use a projected coordinate system, rather than a geographic coordinate system (GCS). As with spatial joins, it is not enough to set the data frame coordinate system to a projection. The tools operate on the saved feature class on disk, and the saved coordinate system is the one being used. When large areas such as continents or large countries are being analyzed, no projection can preserve both area and distance, and one must then choose the projection based on the type of analysis being done.

Often a GCS is employed by data providers because they cannot predict what coordinate systems the user will need. When downloading data, it will often be in a GCS, and the user must choose an appropriate coordinate system and project the data to it. Chapter 3 explains this process. When you are setting up a database for a mapping and analysis project, a suitable projection should be chosen based on the scale and the extent of the data being analyzed. It must be kept in mind that distortions present in the map projection, if significant, will affect distance and area measurements. It remains important to choose a suitable map projection with minimal distortion of area and distance. Review the coordinate system properties on the inside front cover and the coordinate system selection guidelines in Chapter 3, if necessary.

Areas and lengths of features

Geodatabases and coverages automatically create and update fields containing the areas and perimeters of polygons or the lengths of lines. Shapefiles do not maintain this information. Users may create AREA or LENGTH fields for shapefiles and calculate the values manually. However, be careful when using information from these fields. If polygons in a shapefile are clipped, dissolved, or intersected or they undergo any other operation that changes their shape, the AREA fields will not be updated automatically. The user must manually update the fields again to ensure they are correct. Be cautious using an AREA or a PERIMETER or a LENGTH field in a shapefile unless you are sure they are correct. Chapter 6 introduced the Calculate Geometry tool for updating areas, lengths, and perimeters in a variety of units.

Summary

- Map overlay resembles a spatial join, but it splits features when they partly overlap. This function enforces a one-to-one relationship between features when their attributes are joined in the output table.

- Map overlay comes in two basic types. A union keeps all the features from both layers. An intersect keeps all the features that are common to both input layers. Attributes from both layers are joined together in the output.

- A dissolve combines features within a data layer if they share the same attribute. This function can be used to convert many street segments into a single line feature or to remove boundaries between parcels with the same zoning.

- Buffers are polygon constructions that enclose the area within a certain distance of features. Buffers may be created for points, lines, or polygons, and they may be constructed as single or multiple rings.

- Append and merge allow feature classes with the same feature type to be combined as a single feature class, such as merging two adjacent quadrangles to make a single file.

- Map overlay and spatial analysis are generally best done using a projected coordinate system, especially if one anticipates determining areas and lengths as part of the analysis.

- Geoprocessing involves stringing together a sequence of commands during spatial analysis. Spatial functions may be executed from the menus, ArcToolbox, the command line, Model Builder, or scripts.

- The Shape_Area, Shape_Length, and Shape_Perimeter fields in geodatabase feature classes are stored and updated automatically. Other area-based or length-based fields, such as Area, Acres, or Road_km, are not automatically updated.

- Lengths and areas of features stored in shapefiles must be updated manually.

Important Terms

append	Environment settings	merge	suitability analysis
ArcToolbox	erase	Model Builder	union
buffer	geoprocessing	parameter	XY tolerance
clip	intersect	script	
dissolve	map overlay	sliver	

Chapter Review Questions

1. Which different types of outputs are possible (points, lines, polygons) when performing intersect and union?
2. What is the most important difference between a spatial join and a map overlay?
3. What are slivers? Explain how they can be prevented.
4. What is a buffer? Why is a dissolve often performed when buffering features?
5. What function would you use to create a map of a study area such that all the features in the map stopped at the study area boundary?
6. What attribute fields will be present in a layer resulting from a dissolve?
7. Why is it usually advantageous to use a projected coordinate system when doing a map overlay?
8. How can you determine the areas of polygons in a geodatabase? In a shapefile?
9. What determines the coordinate system of the output when overlay is used?
10. What is geoprocessing? In what different ways can commands be executed?

Mastering the Skills

Teaching Tutorial

The following examples provide step-by-step instructions for doing basic tasks and solving basic problems in ArcGIS. The steps you need to do are highlighted with an arrow ➔; follow them carefully. Click on the video number in the Video Index to view a demonstration of the steps.

To demonstrate map overlays, we will do the problem described earlier in the chapter concerning delineating habitat for the rare Black Hills snail using overlays. This habitat will be defined by three criteria: on a limestone geology unit, in dense coniferous forest, and between the elevations of 1200 and 1600 meters.

➔ Start ArcMap and open the map document ex_10.mxd.

➔ Use Save As to rename the map document. Remember to save it often as you work.

TIP: You were directed to turn off background processing in Chapter 2, but choose Geoprocessing > Geoprocessing Options from the main menu now to be sure it is disabled.

As in Chapter 9, we don't want the new feature classes created during the analysis to become part of the original geodatabase, so we'll create another geodatabase to contain them. We'll also set the geoprocessing Current Workspace as the default location to place output data.

1➔ Click the ArcToolbox tab, or the button on the main toolbar to open it. Dock it on the right side of the ArcMap window, if necessary.

2➔ In ArcToolbox, expand the Data Management > Workspace entries and double-click on the Create File GDB tool.

2➔ Click the Browse button for the Location and navigate to the mgisdata folder. Select the BlackHills folder by clicking it once (don't go inside it) and click Add.

2➔ Enter **chap10results** for the output geodatabase name. Click OK.

3➔ From the main menu bar, choose Geoprocessing > Environments.

3➔ Expand the Workspace entry and use the Browse button to set the Current Workspace to chap10results (Fig. 10.16).

3➔ Also set the Scratch Workspace to chap10results.

3➔ Expand the Output Coordinates entry and check that the Output Coordinate System is set to Same as Input. Click OK.

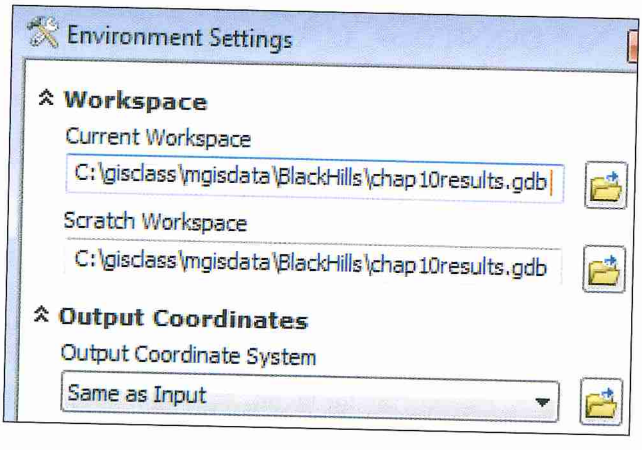

Fig. 10.16. Setting geoprocessing environments

Preparing to overlay

Our first step is to create layers that contain polygons where each condition holds. We will begin with the Geology layer. The limestone areas include the Madison Formation and the Upper Paleozoic units. We will use an attribute query to place these units in a separate layer.

4➔ Choose Selection > Select By Attributes from the main menu bar.

4➔ Set the layer to Geology and enter the expression:

NAME = 'Upper Paleozoic' OR NAME = 'Madison Limestone'. Click OK.

4➔ Right-click the Geology layer and choose Selection > Create Layer from Selected features.

4➔ The new layer appears at the top of the Table of Contents. Rename it Limestone.

TIP: Creating a layer from a selection, although not necessary for processing, is often a good practice. It keeps the selection available if a question about the procedure arises later, or if a tool must be rerun due to an error.

Next, we will select the dense conifers and create a layer from them. We will do the query in two steps. First we will select the ponderosa pine (TPP) and white spruce (TWS). Then we will select the dense areas (class contains C or 5) from the already selected set.

5➔ Turn on the Vegetation layer and turn off the Limestone layer.

5➔ Clear all selected features.

5➔ Open the Select By Attributes window again and set the layer to Vegetation.

5➔ To select only the conifers, enter the expression:

COV_TYPE = 'TPP' OR COV_TYPE = 'TWS'. Click Apply.

6➔ Change the selection method to *select from current selection.*

6➔ Clear the expression and enter another one that says:

DENSITY96 = 'C'. Verify the expression and click OK.

6➔ Right-click the Vegetation layer and choose Selection > Create Layer from Selected Features.

6➔ Rename the new layer Dense Conifer. It should look like the polygons shown in Figure 10.17.

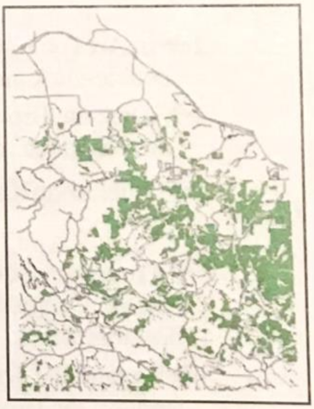

Fig. 10.17. Dense conifer

The elevation range has already been prepared. You are almost ready to intersect.

7➔ Clear the selected features and turn off the Vegetation layer.

7➔ Turn off all layers except the Elevation Range layer, the Limestone layer, and the Dense Conifer layer.

7➔ Zoom in to the middle right of the map and examine the Dense Conifer polygons.

Notice that these polygons have many boundaries inside them because they are divided based on age and density as well as species. The amount of time needed to intersect a layer is proportional to the number of features in the layer. To streamline the intersect, we are going to remove the unnecessary boundaries between these polygons using the Dissolve tool.

8➔ In ArcToolbox, expand the Data Management > Generalization entries and double-click the Dissolve tool to start it (Fig. 10.18).

When using a tool for the first time, it is wise to select Show Help to find out what the tool does and what the **parameters** mean.

8➔ Click Show Help on the Dissolve tool and read the description.

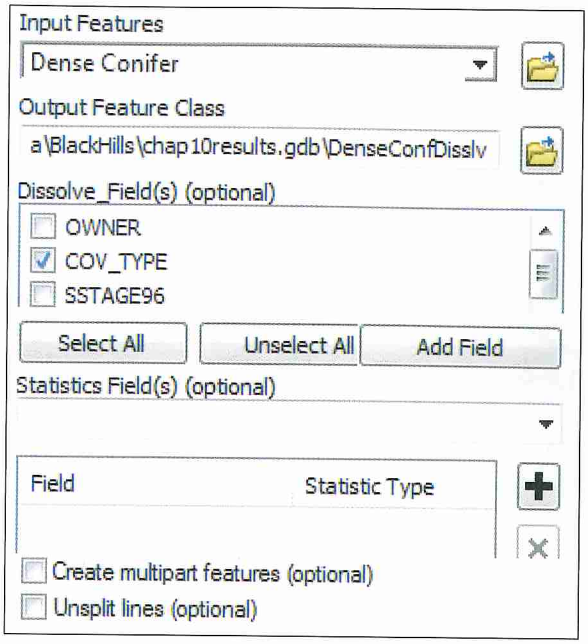

Fig. 10.18. The Dissolve tool

8➔ Click on the Input Features box. The Help message changes to describe the Input Features parameter.

8➔ Click on and read the descriptions for the other input parameters in the window.

8➔ To bring the tool description back, click on the gray tool area.

8➔ Click on the Tool Help icon at the bottom of the tool for more detailed information. Read about the Dissolve tool.

8➔ Close the ArcGIS Desktop Help window and click Hide Help on the Dissolve tool.

9➔ Click the drop-down arrow in the Input Features box and set it to Dense Conifer (Fig. 10.18).

9➔ Click the Browse button to place the Output Feature Class in the BlackHills/chap10results geodatabase and name it DensConifDisslv.

9➔ Check the box to dissolve on the COV_TYPE field.

9➔ Keep the box unchecked to *Create multipart features*. We want unconnected polygons to remain separate features.

9➔ Notice the other options in the tool. We don't need to change any of them, however.

9➔ Click OK.

TIP: When using a tool, be aware that some options may be out of sight, and you must scroll down to see them. Often the defaults are fine, but get in the habit of checking before you run it.

10➔ Examine the output file and note that the intervening boundaries have disappeared.

10➔ Right-click the Dense Conifer layer and choose Remove.

10➔ Rename the DensConifDisslv layer Dense Conifer.

Intersecting polygons

Now we are ready to overlay. We will use Intersect because we want to find the areas common to all three layers. Because the tool can intersect only two layers at a time, we must use it twice.

11➔ Open the ArcToolbox > Analysis Tools > Overlay > Intersect tool (Fig. 10.19).

11➔ Click Show Help and examine the entries.

1. What are the options for the Join Attributes parameter? Which one do you think is best to use?

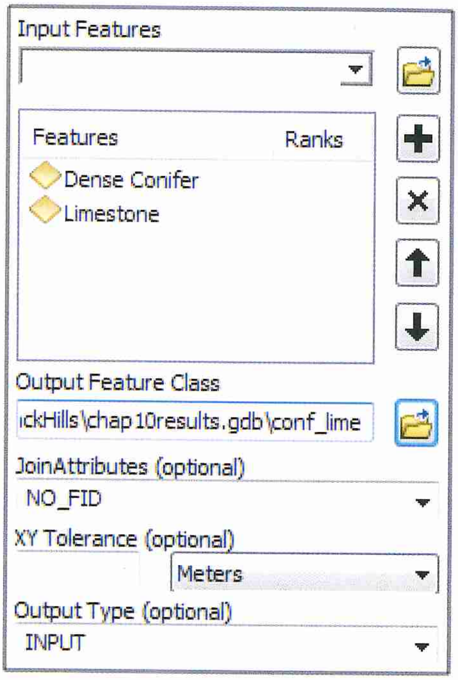

Fig. 10.19. The Intersect tool

11➔ Click on the drop-down button under Input Features and choose the Dense Conifer layer. It will be added to the list of Features.

11➔ Click the drop-down button again to add the Limestone layer to the list.

11➔ Enter the Output Feature Class, placing it in chap10results and naming it conf_lime.

11➔ Set the Join Attributes option to NO_FID.

11➔ The remaining options can be left with their defaults. Click OK.

11➔ Examine the output. The conifers outside the limestone have disappeared.

TIP: The Feature ID, or FID, is a unique integer assigned to every feature for tracking. During an intersection, new FID values are assigned to each new feature. The NO_FID option eliminates the original FID values from the output table, since they have little use.

12➔ Use the Intersect tool again to intersect conf_lime and the Elevation Range layers, naming the output snailhab and putting it in chap10results.

TIP: If you have an ArcGIS Advanced license, you can intersect more than two layers at a time.

13➔ Zoom out to the full extent of the data.

13➔ Rename the snailhab layer Snail Habitat.

13➔ Remove the Limestone, Dense Conifer, and conf_lime layers.

13➔ Zoom in to the extent of the Snail Habitat and turn on the Roads layer.

Overlay of lines in polygons

The snails have a three-week breeding season in early June. During this period they seek the open areas offered by roads, and many get crushed. The Forest Service wants to consider closing primitive roads that traverse through snail habitat during the breeding season to lessen the number of crushed snails. They need to assess which roads must be closed. You will make a map with the proposed road closures. This process involves a line-in-polygon intersection.

14➔ Use Select By Attributes to select the Primitive roads TYPE = 'PR'. (Make sure you set the method back to *Create a new selection.*)

15➔ Intersect the Roads layer (with the primitive road selection) and the Snail Habitat layer. Name the output propclose and put it in chap10results.

15➔ Rename the propclose layer Proposed Closures. Clear the selected features.

16➔ Create a map similar to Figure 10.20, with the closure roads highlighted in red and the other roads in black. Turn off the Elevation Range layer. Set the transparency of the Snail Habitat layer to 50% to help bring out the road patterns.

16➔ Save your map document.

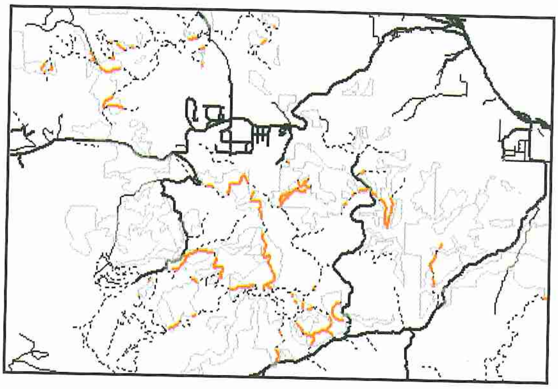

Fig. 10.20. Map showing proposed road closures in snail habitat areas

Clipping layers

The vegetation layer does not extend as far north as the rest of the data, but the map may give readers the false impression that the analysis is valid there. Clipping the Roads layer at the edge of the vegetation layer will prevent any misunderstanding. You need the Clip tool. Recall that you can use the Search window to find tools.

17➔ On the main menu bar, choose Geoprocessing > Search For Tools.

17➔ Type clip in the search box. Several suggestions appear, including Clip (analysis) and Clip (management). The words in parentheses refer to the area of the Toolbox.

17➔ Ignore the suggestions for now and click on the magnifying glass button.

The list includes some exact matches (clip) and some similar functions, such as clipping rasters, along with a description of each.

17➔ Click the Options button in the search window and choose Search Options.

17➔ Click the General tab and check the box to *Show Pop-up window in search results.*

17➔ Click OK and hover the cursor over the blue tool names to see the descriptions. The Clip (Analysis) appears to be the one to use.

17➔ Click the blue Clip (Analysis) text to open the tool.

TIP: You can pin the Search window to the right tab bar for easy access later.

18➔ Set the Input Features to Roads. Set Vegetation as the Clip Features.

18➔ Store the output in the chap10results geodatabase as roadclip. Click OK.

18➔ Turn off the Roads layer.

You can quickly transfer the symbology of the old Roads layer to the clipped roads.

19➔ Open the properties of the roadclip layer and click the Symbology tab.

19➔ Click the Import button and choose to import symbology from a layer or layer file.

19➔ Click the drop-down box to set the layer to Roads. Click OK.

19➔ The road classification is based on the TYPE field. The clipped roads use the same field, so leave the TYPE name in the box. Click OK and OK.

20➔ Rename the roadclip layer Clipped Roads.

20➔ Click and drag the Clipped Roads layer below the Snail Habitat layer.

Working with buffers

To prevent snail death on the major roads, which cannot be closed, a Forest Service biologist has suggested thinning the tree stands within 200 meters of the roads to make these areas less attractive to snails and keep them away from roads. We can prepare a map showing the stands to be cleared and determine the total percentage of snail habitat that would be eliminated.

21➔ Use Select By Attributes to select the primary and secondary roads from the Clipped Roads layer, using the expression TYPE = 'P' OR TYPE = 'S'.

21➔ Create a layer from the selected roads and name it Major Roads. Clear the selected features.

TIP: Buffers are, time-consuming to create, so it is helpful to reduce the number of features before buffering. We will use the Elevation Range layer to clip the roads before buffering.

22➔ Open the tool ArcToolbox > Analysis > Extract > Clip.

22➔ Set Major Roads as the Input Features and Elevation Range as the Clip Features.

22➔ Save the result in chap10results as majroadclip. Click OK to run the tool.

22➔ Turn off the other road layers to examine the new file.

Now buffer the clipped roads.

23➔ Open the tool ArcToolbox > Analysis > Proximity > Buffer.

23➔ Set the Input Features to majroadclip and the Output Feature Class to roadbuf in chap10results.

23➔ Set the Linear unit to meters and type 200 in the box.

23➔ Scroll down, if necessary, and set the Dissolve Type to ALL. Click OK.

TIP: By default, the Buffer tool does not dissolve boundaries between buffers. This is one case where the default option is not right, and the Dissolve Type must be set to ALL.

Now intersect the road buffers with the snail habitat to determine the potential areas to be cleared.

24➔ Open ArcToolbox > Analysis > Overlay > Intersect.

24➔ Choose Snail Habitat and roadbuf as the Input features.

24➔ Specify the Output file as proposedthin in chap10results. Click OK.

25➔ Clean up the map display to show the results. Turn off the roadbuf layer and make the proposedthin layer a symbol that contrasts well with the snail habitat (Fig. 10.21).

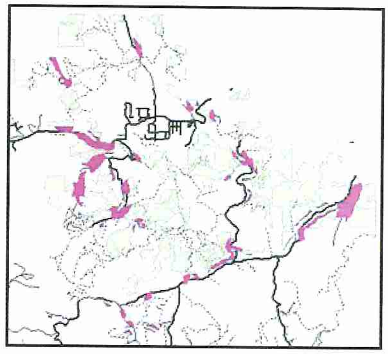

Fig. 10.21. Proposed thinning areas (in pink)

Now the final step is to determine the area of snail habitat that will be eliminated by the thinning. Geodatabases automatically keep track of the areas, lengths, and perimeters of features.

26➔ Open the proposedthin table and examine the Shape_Area and the Shape_Length fields.

These fields were renamed Shape_1_Area and Shape_1_Length during one of the intersections, but they will contain the correct areas and perimeters for the feature class. The units of the Shape_Area field will be in the same coordinates as the map units of the feature class coordinate system, in this case, square meters.

26➔ Use the Statistics function to determine the total area to be thinned.

TIP: To convert square meters into square kilometers, divide by 1 million (1,000,000).

2. What is the total area of the proposed thinning in square kilometers? ________________
3. What is the total area of snail habitat in square kilometers? ______________________
4. What percentage of the habitat would be eliminated by the proposed thinning? ______

TIP: Shapefiles do not automatically maintain area, length, or perimeter fields. Values in these fields may not be correct if geoprocessing actions have been performed since the fields were created. The fields may be updated using the Calculate Geometry tool described in Chapter 6.

Investigating relationships with union

Intersect is valuable for finding areas where known criteria exist simultaneously, such as finding the snail habitat based on known ideal factors of elevation, geology, and vegetation. Union is a powerful tool for investigating relationships between two types of information.

It is well known that rock units have an impact on the soils that are produced from them. It would be interesting to see if the soils then have an impact on the productivity of the forest. We will perform a union on the Geology and Vegetation layers to explore this question.

27➔ Turn off all layers except Geology and Vegetation.

27➔ Open ArcToolbox > Analysis > Overlay > Union.

27➔ Choose Geology and Vegetation as the Input Features.

27➔ Name the output feature class geolveg and put it in the chap10results geodatabase. Click OK.

28➔ Zoom to the full extent and turn off all layers except for geolveg.

First, notice that all areas of both input layers are included in the output, even though the Vegetation layer does not cover as much area as the Geology layer does.

28➔ Click the Identify tool. Click on several of the large polygons at the top and examine the attributes.

28➔ Click on several small polygons at the bottom.

Most of the smaller polygons occur where both Vegetation and Geology were present. Their attributes are stocked with information from both layers (Fig. 10.22). However, the polygons at the top came only from Geology, so the vegetation fields are full of blanks and zeroes.

Field	Value
OBJECTID	135
Shape	Polygon
FID_geology	15
UNIT	5
AGE_NUM	4
NAME	Upper Paleozoic
INFIL	Mod
INFILCLASS	2
FID_vegetation	241
RISDATA	0814130007
DATA	0814130007
OWNER	NFS
COV_TYPE	TPP
SSTAGE96	4B
TREE_SZ96	L
DENSITY96	B
SITE_NDX	65

Fig. 10.22. Attributes from union of geology and vegetation

To investigate the relationship between geology and site productivity, we will Summarize on the Geology NAME field, and request statistics from the SITE_PROD and CROWN_COV fields, both indicators of forest productivity. However, because some polygons have no vegetation data, we will exclude the zero values from consideration with a query.

29➔ Close the Identify window and open the geolveg table.

29➔ Open the Table Options > Select By Attributes window and enter the expression SITE_PROD > 0. Click Apply and close the query window.

30➔ Right-click the NAME field and choose Summarize.

30➔ Expand the SITE_PROD field and check the boxes for Minimum, Maximum, Average, and Standard Deviation.

30➔ Expand the CROWN_COV field and check the boxes for Minimum, Maximum, Average, and Standard Deviation.

30➔ Name the output table geolproductivity and save it in the chap10results geodatabase. Click OK and add the table to the map.

31➔ Open the geolproductivity table. Find the Average_SITE_PROD field and use Sort Descending. Examine the averages and standard deviations.

31➔ Find the Average_CROWN_COV field and use Sort Descending. Examine the averages and standard deviations.

Although both site productivity and crown cover vary from unit to unit of the geology, for the most part all the averages are within one standard deviation of the others. This analysis does not support the hypothesis that the geological units have a significant impact on forest productivity.

31➔ Close the Table window and clear all selected features.

Working with slivers and tolerances

We will now work on a short exercise to demonstrate the problem of slivers and how to help prevent them using tolerances. We need a different data frame for this part of the tutorial.

32➔ Choose Insert > Data Frame from the main menu bar. Collapse the Sturgis Area data frame to hide the layers you've been working with.

32➔ Add the cd111 and counties feature classes from the mgisdata\Usa\usdata geodatabase.

32➔ Zoom in to the San Francisco Bay Area in northern California (Fig. 10.23).

32➔ Choose Bookmarks > Create Bookmark. Enter the name San Francisco and click OK.


Fig. 10.23. Zoom in to this area.

This area will be our experiment. We don't need to overlay the entire country, so to save processing time, we will use the Environment settings to constrain the output extent to this display window.

33➔ From the main menu bar, choose Geoprocessing > Environments.

33➔ Expand the Processing Extent entry.

33➔ Change the Extent drop-down to read Same as Display. Click OK.

34➔ Open the tool ArcToolbox > Analysis > Overlay > Union.

34➔ Enter cd111 and counties as the input features.

34➔ Name the output union_notol and put it in chap10results. Click OK.

TIP: Once set to an extent, the Processing Extent setting will continue to use the same range of *x-y* coordinates, so it will not be affected as you zoom in and out.

35➔ Zoom to the full extent of the data.

35➔ Notice that only a small group of polygons around San Francisco is included in the output. This is the result of the Processing Extent environment setting.

35➔ Choose Bookmarks > San Francisco, and examine the output.

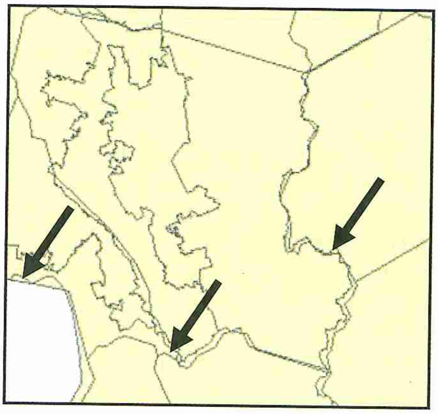
Fig. 10.24. Slivers from the union

Notice that many slivers occur (Fig. 10.24). The district and county boundaries coincide in real life, but these two data sets come from different sources, and the boundaries aren't identical. To specify a processing tolerance, it would be helpful to know the typical width of the slivers.

36➔ Zoom in to a region with typical slivers.

36➔ Click the Measure tool on the Tools toolbar. Set the units to meters.

36➔ Measure across several slivers to estimate their widths. Close the Measure tool.

You will find that a typical width is about 1000 meters. A tolerance is a radius, so we should use about half the measured value. Let's repeat the union and use a tolerance of 500 meters.

37➔ Open the ArcToolbox > Analysis > Overlay > Union tool.

37➔ Set the input layers to cd111 and counties, as before.

37➔ Name the output union500 and place it in the chap10results geodatabase.

37➔ Enter 500 meters as the XY tolerance. Click OK and examine the output.

37➔ Turn the new layer on and off a few times to compare the boundaries with the union_notol layer. Many slivers are gone in the new layer.

Notice two things about the new layer. First, many slivers are still present. Our initial tolerance was not sufficient to remove them. Second, the district boundaries have lost some of their detail. The XY tolerance may not have fixed all the slivers, but it already is having an effect on the accuracy of the output. Nevertheless, we will try a larger tolerance.

38➔ Repeat the union of cd111 and counties, but this time use an XY tolerance of 1000 meters and name the output union1000.

38➔ Turn on just the union_notol and the union1000 layers and compare them by turning the top one on and off a few times.

Most of the slivers are now gone, but the loss of detail in the district boundaries is apparent. The county boundaries don't look much worse, but they were a lower resolution to begin with.

39➔ Try one more union with a 1500-meter tolerance and compare.

Now the generalization of the boundaries is obvious, and a few stubborn slivers still remain. Further generalization and loss of accuracy are not worth getting rid of the few remaining slivers. The 1000-meter tolerance gave the best compromise performance. Before we go on, we must set the Environment settings back to their defaults.

40➔ Choose Geoprocessing > Environments from the main menu bar.

40➔ Expand the Processing Extent entry and set it back to Default. Click OK.

TIP: One challenge of using Environment settings is to remember to reset them after you are finished. Note that the Environment settings are saved with map documents.

Geoprocessing with Model Builder (optional)

This section shows how to use Model Builder to record sequences of geoprocessing steps. Since the snail habitat problem is familiar, we'll use it as our example.

➔ Open the original version of ex_10.mxd without any changes.

➔ Use Save As to save it under a new name and save often as you work.

Model Builder often creates intermediate data sets while it works. You can specify a scratch workspace where these are placed, so that they don't interfere with the original data. You set the workspace using the Environment settings.

41➔ Open Geoprocessing > Environments from the main menu bar.

41➔ Expand the Workspace entry. Set both the Current Workspace and Scratch Workspace to the chap10results geodatabase. Click OK.

5. You already set these Environments earlier in the tutorial. Why do you have to set them a second time now?

Creating and running a model

You store a model inside a geodatabase or in a toolbox. We will create a toolbox for it.

42➔ Click the Model Builder button on the main tool bar.

42➔ Choose Model > Save in the Model Builder window.

42➔ Navigate so that you are in the BlackHills folder.

42➔ Click the New Toolbox button in the Save window.

42➔ Name the toolbox BHtools and click Enter.

42➔ Double-click the new BHtools toolbox to enter it.

42➔ Type SnailHab as the name for the model and click Save.

43➔ Choose Model > Model Properties from the Model Builder menu bar.

43➔ Label the model Snail Habitat Analysis and give it a description (Fig. 10.25).

43➔ Examine the other settings as shown in Figure 10.25 and then click OK.

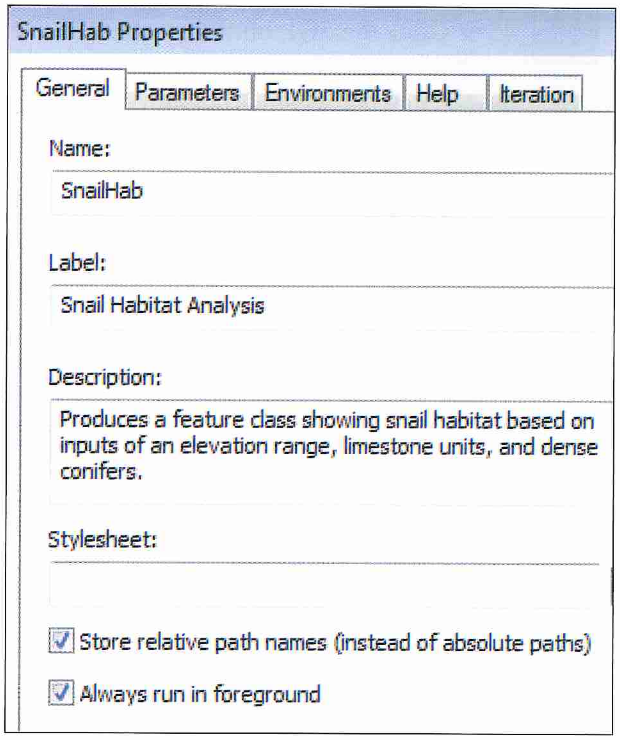

Fig. 10.25. Naming and describing a model

TIP: Storing relative pathnames for a model makes it work if the model and data together are moved to a different folder. Absolute pathnames only work if the data stays put.

In Model Builder, you paste the desired tool in the window, set arguments for the tool, and run it. Each step can be run separately as the model is built or all together. When building the model, it is helpful to execute the steps one at a time. In the beginning of this tutorial, we used Select By

Attributes to select the desired polygons from the geology and vegetation layers. In Model Builder, we use the Make Feature Layer tool.

44➔ Find the ArcToolbox > Data Management Tools > Layers and Table Views > Make Feature Layer tool. Click and drag it to the Model Builder window.

44➔ Right-click the Make Feature Layer tool in the Model Builder window and choose Open.

44➔ Set the Input Features to Geology and name the Output Layer Limestone, as shown in Figure 10.26.

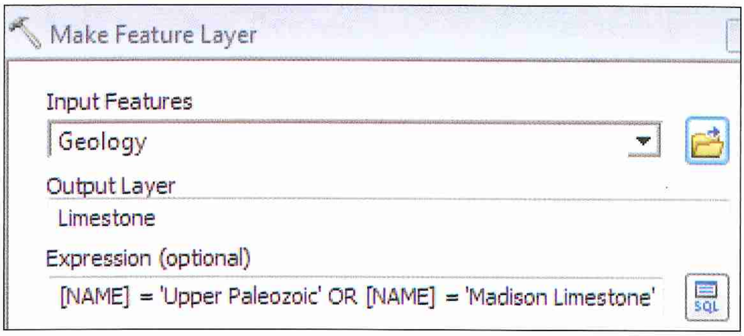

Fig. 10.26. Setting the arguments for the Make Feature Layer tool to create the Limestone layer

45➔ Click the SQL button to enter the selection expression: [NAME] = 'Upper Paleozoic' OR [NAME] = 'Madison Limestone'. Verify the expression.

45➔ Click OK to close the SQL window and OK to close the tool.

45➔ The boxes in the model become colored, indicating the tool is ready to run. Choose Model > Run, or click the Run button to execute this part of the model.

45➔ Close the dialog box after the tool completes successfully. A drop shadow behind the model shapes indicates that the tool has been run (Fig. 10.27).

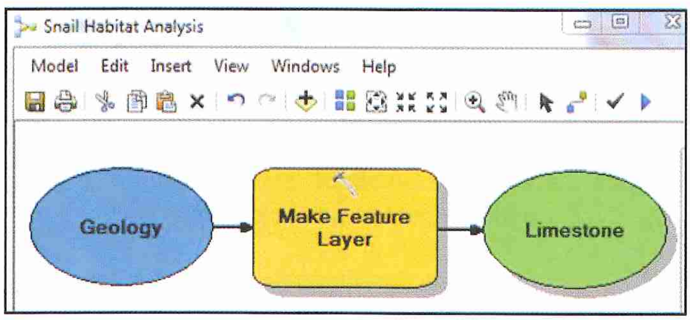

Fig. 10.27. The model after the first step is run

The blue ovals indicate inputs, and the green ovals indicate outputs. Both inputs and outputs have properties that can be set.

45➔ Right-click the green Limestone oval and choose Add to Display. The new layer appears in the Table of Contents and is shown on the map.

Now we'll use the same tool to select the dense conifers. Because we're using a tool, it is easier to do the selection in one step.

46➔ From ArcToolbox, click and drag another copy of the Make Feature Layer tool onto the model canvas.

46➔ Right-click the tool to open it. Set the Input Features to Vegetation and the Output Layer to Dense Conifer.

46➔ Click the SQL button and carefully enter the expression (with the parentheses): ([COV_TYPE] = 'TPP' OR [COV_TYPE] = 'TWS') AND [DENSITY96] = 'C'.

46➔ Click the Verify button to ensure that you did not make a mistake in the query.

47➔ Click OK to close the SQL window and OK to close the tool window.

47➔ Click the Run button and close the dialog window after it finishes.

47➔ Right-click the Dense Conifers output and choose Add to Display. Make sure that the selection looks correct (see Fig. 10.17).

TIP: Click on a model shape to select it. Use click and drag boxes or the Shift key to select multiple shapes. Click and drag selected shapes to move them around on the model canvas or to resize them. Use these techniques to arrange the model in neat patterns.

The Auto Layout button can be used to quickly arrange the model in a logical pattern. 

48➔ Click and drag the ArcToolbox > Analysis > Overlay > Intersect tool onto the model canvas.

48➔ Right-click the Intersect tool on the model canvas and open it.

48➔ Set the Input Features to Limestone and Dense Conifer. When choosing the layers, be sure to choose the ones with the blue symbol, indicating that they are outputs from the model.

48➔ Name the Output Feature Class conf_lime and store it in the chap10results geodatabase. Click Yes to overwrite the one that exists, if you are asked.

48➔ The tool should look like Figure 10.28. Leave the other arguments set to their defaults.

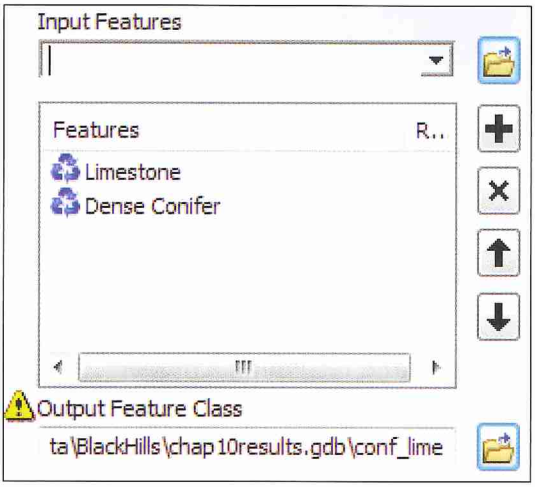

Fig. 10.28. Setting the arguments for the first Intersect tool

Notice the red or yellow warning sign next to the Output Feature Class. Put the cursor on it to read the message that says that this output already exists. Of course it does, since we've done this analysis before. When running models, you generally want to be able to overwrite previous results. We can set this option.

49➔ Click OK to close the tool.

49➔ Choose Geoprocessing > Geoprocessing Options from the main menu bar.

49➔ Check the box to *Overwrite the results of geoprocessing operations*. Click OK.

50➔ Drag another copy of the Intersect tool from ArcToolbox to the model canvas.

50➔ Open the second Intersect tool.

50➔ Select the conf_lime model output from the Input Features drop-down box.

50➔ Choose Elevation Range from the Input Features drop-down box.

50➔ Name the output feature class snailhab2 and place it in chap10results.

50➔ Leave the rest of the arguments to their defaults and click OK.

51➔ Choose Model > Save to save the model in its current form. It should look similar to the model in Figure 10.29.

51➔ Run the model. The two intersections you set up will both be executed.

51➔ Right-click the snailhab2 output to add it to the display and examine it.

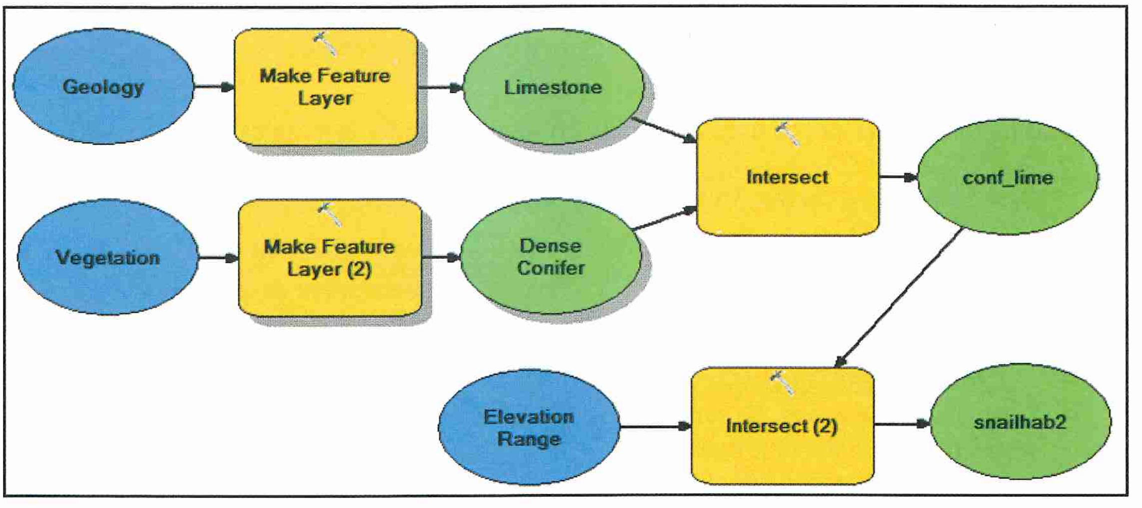

Fig. 10.29. The completed Snail Habitat model

Changing and rerunning a model

The saved model provides a record of the inputs and parameters used to derive the output. It can also be edited and run again to try out different solutions. Imagine that a geologist colleague has commented that the Lower Paleozoic geological units also contain limestone and should be included with the other geology layers.

52➔ Right-click the Make Feature Layer input box for the Geology query and open it.

52➔ Click the SQL button and click at the end of the current expression to place the cursor there.

52➔ Add the rest of the expression, OR [NAME] = 'Lower Paleozoic', so that all three units will be selected. Verify the query and then click OK to close the tool.

Notice that the model boxes lose their shadows (except the unchanged Vegetation query), indicating that they need to be run again with the new conditions.

53➔ Right-click the Make Feature Layer tool from the Geology query again and choose Run to run only this step of the model to check your query.

53➔ The previous Limestone output was removed from the display. Right-click its green oval in the model and choose Add to Display to uncheck it. Repeat to add it to the display a second time. Examine it.

54➔ Click the Run button in the Model Builder toolbar to run the rest of the model steps.

54➔ Add the snailhab2 output to the map and examine it to see how it changed.

54➔ Save the new version of the model by choosing Model > Save.

Creating a tool from a model

Finally, models can be set up to run as tools, by turning the inputs into **parameters**. This approach is helpful when the same series of steps is to be run with different inputs each time.

55➔ Right-click the blue Vegetation oval and choose Model Parameter. A small P appears to the right of the blue oval, indicating that it is now a parameter.

55➔ Right-click on the snailhab2 green oval and make it a parameter.

55➔ Save the model and close it.

Before we can run the tool, we must add your toolbox to ArcToolbox.

56➔ Open ArcToolbox. Right-click the ArcToolbox entry at the top and choose Add Toolbox.

56➔ Navigate to the BlackHills folder and click on the BHtools toolbox. Click Open.

56➔ Open ArcToolbox again and double-click the BHtools > Snail Habitat Analysis tool.

The model is opened as a tool, with input boxes for the two parameters. Examine the titles above the boxes, Vegetation and snailhab2. The titles are taken from the names assigned to the ovals in the model. Let's rename them so they make more sense to a user.

57➔ Close the tool. Right-click on the model in the toolbox and choose Edit.

57➔ Right-click the Vegetation oval and choose Rename. Call it Input Vegetation Layer. (You can slightly enlarge the box so that the text fits.)

57➔ Rename the snailhab2 oval Output Habitat Layer.

57➔ Save the model and close the model canvas.

57➔ In ArcToolbox, double-click on the model to open it as a tool.

TIP: Double-clicking or choosing Open will open the model as a tool to be run. To open the model canvas, right-click the model and choose Edit.

There is a more recent vegetation feature class in the Sturgis83 geodatabase called veg2006. We'll run the model again using the new data to see if the habitat has changed in the last decade.

58➔ Click the Browse button and find the veg2006 feature class in the Sturgis83 geodatabase. Add it.

58➔ Set the output feature class to snailhab06 and put it in the chap10results geodatabase. Click OK to run the tool.

58➔ Turn off all layers except the Roads, snailhab2, and snailhab06.

58➔ Compare the snailhab2 and snailhab06 layers, noting the change in habitat.

Recall that to create the tool we designated the vegetation input and the habitat output as parameters that could be set each time the tool is run. You can also set individual arguments as parameters. Imagine that we wanted to be able to select different types of vegetation each time we ran the tool.

59➔ Right-click the Snail Habitat tool in ArcToolbox and choose Edit.

59➔ Right-click on the Make Feature Layer box associated with the vegetation query and choose Make Variable > From Parameter > Expression. A new blue input oval appears for the expression.

59➔ Rename the new oval Vegetation Query. Move or resize as needed.

59➔ Right-click the Vegetation Query oval and choose Model Parameter. A P appears next to it (Fig. 10.30).

59➔ Save the model and close it.

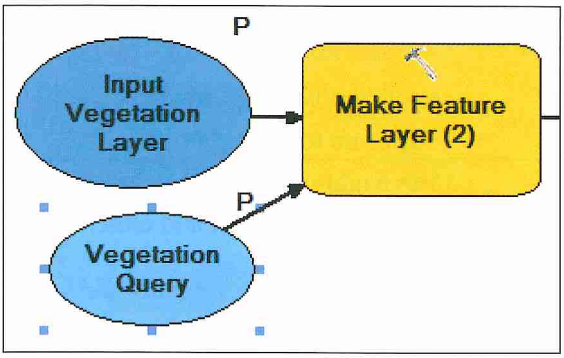

Fig. 10.30. Making the query a parameter

Now let's run the model again. Suppose we want to see the potential habitat if all densely treed areas are considered, not just conifers.

60➔ Open the Snail Habitat model and set the input and output feature classes to Vegetation and snailhab_all.

60➔ Click the SQL button next to the new Vegetation Query input box.

60➔ Clear the existing query and replace it with DENSITY96 = 'C'.

60➔ Click OK to run the tool again.

60➔ Compare the old snailhab2 and new snailhab_all outputs to see the changes in habitat.

This brief introduction shows only a small part of what Model Builder can do. To learn more, read about Model Builder in the ArcGIS Help files.

TIP: Models can be saved as scripts in the Python programming language. If you learn Python, then you can create full-fledged programs with conditional statements (if/then), looping, and so on.

This is the end of the tutorial.

➔ Close ArcMap. Save your changes.

Exercises

Use the data in the BlackHills folder and Sturgis83 geodatabase to answer the following questions. Because solution methods can vary, your answers may be slightly different from the ones in the back of the book.

1. The Madison Limestone is an important aquifer. Citizens are concerned about gas or chemical trucks overturning on roads and causing contamination. What is the total length of roads crossing the Madison, in kilometers?
2. The infiltration rate of the geological substrate is a strong factor in evaluating the spill hazards for roads. Create a map of all the roads based on the infiltration capacity of the geology, using appropriate symbols to indicate hazard. **Capture** your map.
3. The use of a road is another factor in evaluating spill hazards. Create a new field in the roads feature class with a hazard rating of 3 for primary roads, 2 for secondary roads, and 1 for primitive roads. Create a map and **capture** it.
4. Now create new field for a hazard index based on both infiltration and use, adding them together. What is the range of values for the index? Create a map of the roads based on the hazard index. **Capture** the map.
5. Rangeland can cause significant impacts on surface water quality. Using the lulc and wsheds2b feature classes, create a table showing the area of rangeland in each watershed. (Include both herbaceous and mixed rangeland categories.)
6. Areas that are within 300 meters of both streams and primitive roads are the most popular places for fishing. Create a map showing these areas. **Capture** your map.
7. The fishing areas in #6 should be restricted to areas within the national forest (OWNER = 'NFS' in the vegetation feature class). Create a revised map showing the public access fishing areas in green and the private fishing areas in red. **Capture** your map.
8. You need a feature class that shows polygons based only on infiltration and only within the wshds2b area. Symbolize this layer based on infiltration. **Capture** your map.
9. The Forest Service is concerned about septic systems contaminating the Madison Limestone. Create a map showing private land (OWNER = PVT in the Vegetation layer) on the Madison. **Capture** the map.
10. The Forest Service is considering adopting a policy to not cut timber within 200 meters of a primary road (TYPE = P). Assuming that cutting timber includes all SSTAGE96 LIKE '4*', determine the total area in square kilometers lost to harvesting if this policy were adopted. **Capture** a map showing the off-limits areas.

Challenge Problem: A Perfect Place

Imagine that you have always wanted a vacation home in an aspen stand with a little pond. Create a map showing all aspen stands (COV_TYPE = 'TAA') that lie on low infiltration geology units (INFIL = LOW) and are within 500 meters of a primary or secondary road (TYPE = P or S). **Capture** your map.