Visualizing Integrated Three-Dimensional Datasets

Modeling data in the geodatabase using multipatch features

By Alistair Ford

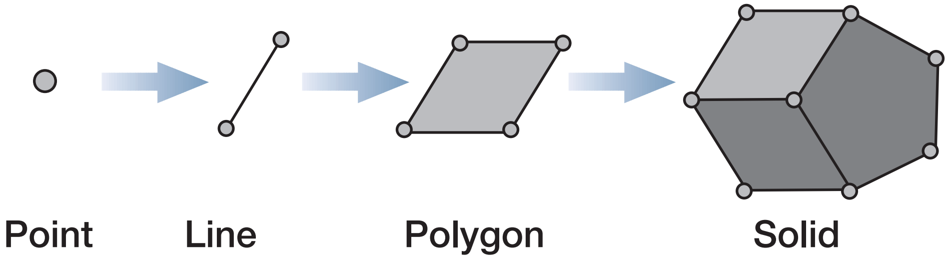
Recent advances in interoperability and standards, in addition to the breakout of GIS from a specialist application toward being a widespread desktop tool, have meant that it is now in an ideal position to provide a powerful data integration tool in many industries. Industries, such as meteorology, environmental management, or oil and gas, deal with complex three-dimensional spatial data. To provide full integration, GIS must handle complex three-dimensional data types and spatial representations that exist in the various disciplines within these industries. This article will examine the current state of three-dimensional data storage in the ESRI suite of products using the oil and gas industry as a case study.

Required Objects and Possible Representations

Many GIS users consider 3D GIS to be a three-dimensional view of standard two-dimensional or two-and-one-half-dimensional spatial data. This three-dimensional aspect is often seen as a cosmetic addition to GIS. Although the data may be visualized in three dimensions, the majority of the spatial analysis, editing, and other common GIS tasks are performed in a strictly two-dimensional environment. While simply viewing data in three dimensions may be sufficient for many applications, some disciplines require a true three-dimensional spatial representation of important objects due to the complex nature of the features to be modeled or to allow for spatial analysis in three dimensions.

Most literature and thought on three-dimensional data modeling for vector GIS present models that are built up from a set of common primitives. Data structures constructed from these primitives are either object oriented or topology oriented, depending on whether the application that will use the data is concerned with visualizations or analysis of the data.

Topological representations expressly store the relationships between any object and its neighbors, while object-oriented models store the structure of the objects, deriving the topology, and therefore spatial relationships as and when required. Because the data integration in this case study is aimed at providing a high-level management tool without replicating any of the specialist analysis present in the originating software, the emphasis should be on the object-oriented representation of features.



Basic three-dimensional geometric primitives

The basic components of any model can be summarized as points (or nodes), lines (or edges), faces (or polygons), and solids. It is quite evident, therefore, that the geometric primitives of a three-dimensional data model are essentially those in a standard two-dimensional or two-and-one-half-dimensional data model.

Point Features

Point features can be represented by the most basic spatial primitive and can be viewed as a three-dimensional location in space. As with two-dimensional GIS, point features rarely exist in the real world as there are very few entities that are zero dimensional (i.e., no area or volume) but are an extremely useful construct for creating higher-dimensional spatial representations. Point features can be used to represent data points, locations where measurements are made or values exist. Examples of data points in the geological case study are geological markers (or well picks), where a boundary exists between two lithological layers along a well.

Line Features

Line features are a collection of point features that are linked together to form a one-dimensional feature. As with point features, line features are abstractions of real-world features that provide suitable representations for simple storage and analysis. Examples of linear features in this case study are well trajectories, pipelines, and cables. In three-dimensional space, a line is still a one-dimensional feature even though it does not sit on a flat surface, as it has only length but not width or height. Point and line features are special because they do not exist in the real world (nothing in the real world has zero width or height) but they are key to constructing more complex three-dimensional features.

Polygon Features

Polygon features are segregated portions of space that define areas with common attributes. A polygon is a two-dimensional feature that consists of one or more rings of lines forming an enclosed boundary. Used to represent planes or surfaces, they are features that have a spatial extent in two dimensions but no thickness. In three-dimensional space, polygons are often viewed as two-and-one-half-dimensional features because the points from which they are built may contain three-dimensional coordinates that give the feature height variation but the feature itself is still two dimensional. Examples of polygonal features in the geological world are fault planes and lithological horizons. A digital elevation model (DEM) can also be viewed as a collection of contiguous polygonal features and is therefore implicitly two-and-one-half dimensional because it is a surface with zero thickness but with height variation.

Solid Features

Solids are “true 3D” features consisting of a number of polygonal faces defining an enclosed boundary within which is a region of common attributes. Unlike the above features, solids are not restricted to two-and-one-half-dimensional representation, which allows complex entities such as oil and gas reservoirs to be represented in a GIS database. However, the creation of solid features is an extremely difficult process. While a polygonal feature can be represented by defining the order of points that create the boundary, determining the order of points that form a solid feature is particularly complex.

Three-Dimensional Feature Storage in ArcGIS

As outlined above, there is a raft of literature that outlines data models for three-dimensional

Continued on page 16

Visualizing Integrated Three-Dimensional Petroleum Datasets

Continued from page 15

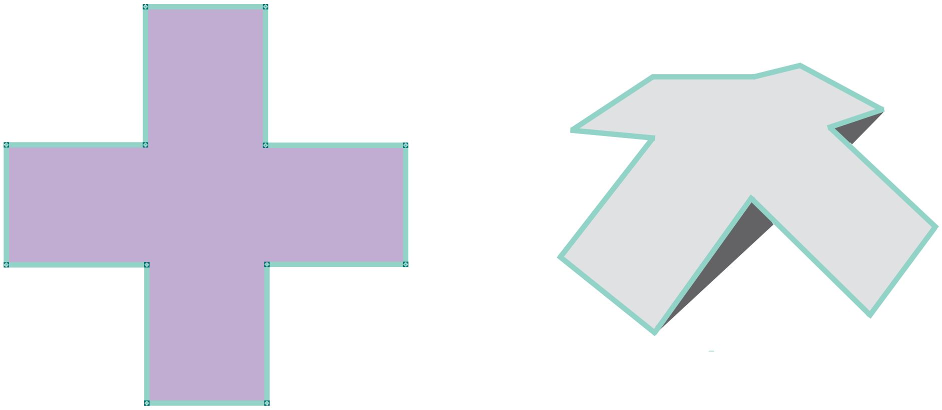
features in GIS databases, although most of this presents excellent theoretical storage methods that could be implemented in bespoke GIS systems. Most large organizations are committed to large commercial GIS and DBMS contracts (e.g., Shell International Exploration and Production with ESRI and Oracle). Any three-dimensional data storage and associated functionality must therefore be developed within the restrictions of the current database and GIS technologies.

Some proposals have been made for the storage of three-dimensional objects in a commercial DBMS (e.g., Oracle), although this results in a solution that is reliant on a single database vendor's technology. In contrast, GIS data storage engines, such as ArcSDE, are designed to be independent of database technologies. Also, ArcSDE allows simpler integration of numerous data types and with spatial data in mind.

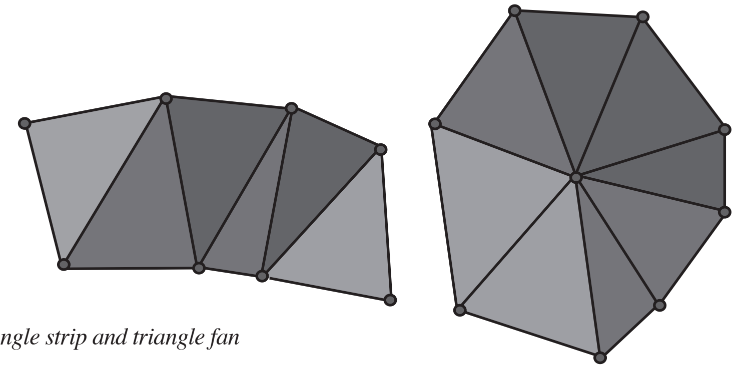
To fully integrate the various surface and subsurface datasets, GIS must be capable of handling true 3D data. While most commercial GIS software (ESRI included) allows the assignment of the third dimension in coordinates and includes some level of three-dimensional visualizations, the features themselves are often limited to two-and-one-half dimensions. This means that while the locations of a feature may contain a z-value, the structure of the feature itself may not be true three dimensional; the feature is restrained to a simple surface. For example, a polygon feature may be constructed from numerous points with z-values, but two points making up the polygon may not have the same (x,y) value. This is because all topological operators acting on the polygon at the time of its creation (to ensure the polygon has no edges that cross, for example) act solely in two dimensions.

One example of this is the attempt to construct a box from a polygonal representation in an ESRI geodatabase. In the real world, a box can be created from a simple net, which itself can be represented in a GIS as a polygon feature. This feature is perfectly valid in two dimensions and even in two-and-one-half dimensions, but once an attempt is made to wrap up the polygon to form the box, the topological operators will remove any duplicate (x,y) points that form the vertical sides of the box (despite the fact that they are distinct vertices with different z-values) as the feature is invalid.

This shows that, in a two-and-one-half-dimensional system, a feature must meet with the demands of a two-dimensional-oriented world. It is obvious from this that the creation of true three-dimensional features using standard geospatial objects such as points, lines, and polygons will not be possible. The provision of a true 3D feature in a geodatabase is required to enable the storage of complex 3D objects alongside the more common two-dimensional or two-and-one-half-dimensional spatial objects.



Polygon box net (shown left) and a 2.5 box net (shown right)



Triangle strip and triangle fan

Multipatch Features

In the ESRI geodatabase, there exists an elusive feature that could achieve such a goal. The multipatch feature is designed for the on-the-fly creation of three-dimensional symbology for visualizations. The multipatch feature is also currently ESRI's answer to true three-dimensional geodatabase objects. It builds on the OpenGL 3D primitives—triangles that can be created in strips and fans. A multipatch feature is simply an additional ESRI geometry type (much like points, polylines, or polygons) and can therefore be assigned in the geometry field of a feature in a shapefile or geodatabase. In this way, spatial entities can be represented with true three-dimensional features and still be queried and selected and have full attribute data and associated symbology within ArcMap and ArcScene.

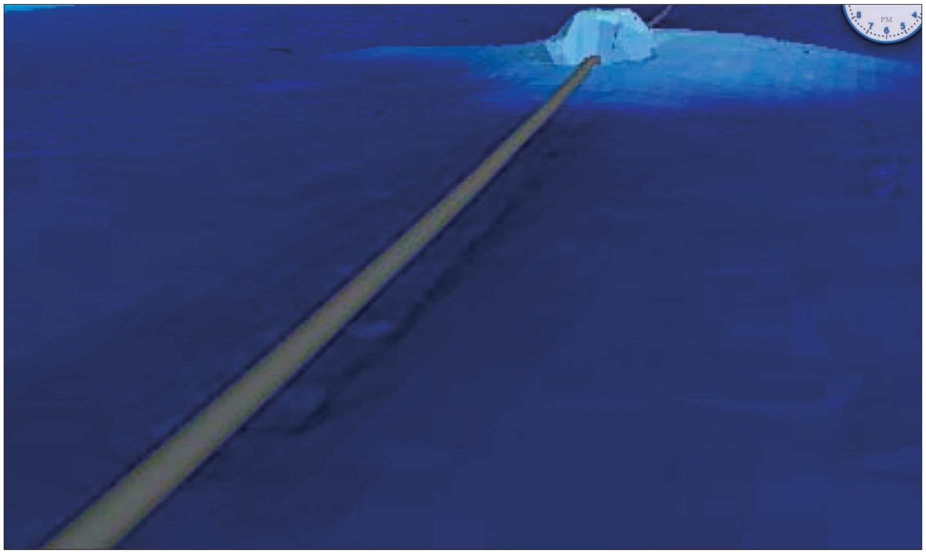
The multipatch is essentially a solid feature, consisting of a number of polygon faces that define a boundary. The multipatch feature can therefore be utilized as a means of storing complex three-dimensional features in a geodatabase. Although multipatch features are specifically designed for display purposes, it can be seen that they are able to fill the role of a true 3D spatial representation.

The multipatch feature cannot be created in normal edit sessions in ArcMap as with the standard ESRI geometry types. These features must be created using ArcObjects, either from existing geometries (how buildings are extruded in ArcScene and ArcGlobe) or from raw data sources. The following section outlines some of the issues with multipatch creation.

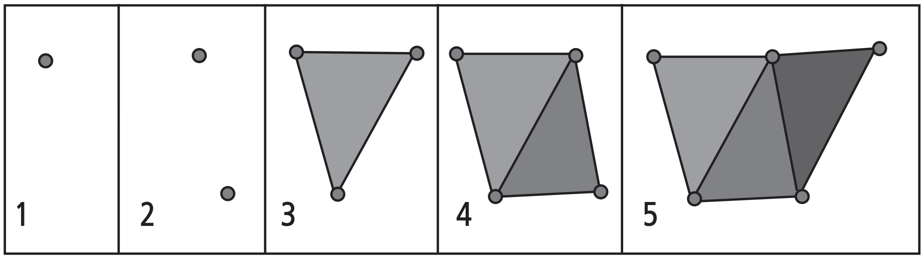
Multipatch Creation Using ArcObjects

After examining a number of potential applications, multipatches were used for storing true three-dimensional features and would be tested on reservoir and pipeline objects. An oil or gas reservoir is a complex three-dimensional object with irregular shape and structure. While the geological interpretation and modeling of such objects were beyond the scope of this study, the storage of the resultant reservoir objects in a geodatabase was seen as an excellent test of the multipatch as a three-dimensional representation.

A pipeline could be stored as a line feature in geodatabases, but for some three-dimensional applications, this representation is insufficient. For example, when carrying out analysis of subsea pipelines for areas where a pipeline on the seabed has become exposed due to ocean currents and is therefore in danger, the inclusion of a true three-dimensional representation can



Three-dimensional pipeline created with a multipatch feature



Triangle strip creation

be combined with a detailed seabed surface (perhaps from a swathe bathymetry survey) to give a visual indication of areas of dangerous exposure.

Since they cannot be created directly in ArcMap, most multipatch features will be created from existing data. In the oil and gas industry, and particularly the geological application area, most data transfer is done in the form of ASCII text files representing features as a series of x,y,z points. These files can then be loaded through ArcCatalog, either as a data table or an XYZ feature class. It is extremely important to know and understand the order of the x,y,z points in these feature classes, as this impinges directly on the way that the multipatch is built up from the raw data.

For example, for data concerning a pipeline, it is reasonable to assume that the x,y,z points will be ordered such that each point is sequenced along the feature. Conversely, with a complex geological feature, knowledge of the structure of the input data is vitally important. Most native data formats used by the modeling and interpretation software in which reservoir models will be created are not open source.

As multipatch features can be assigned to the geometry field of a geodatabase record, their creation through ArcObjects initially follows the same method as creating any standard ESRI feature. Initially the geodatabase feature class must be created, along with the associated spatial reference. Since the multipatch is often created from an input feature class, the spatial reference can be derived directly from this as shown in Listing 1.

```
Dim pGeoDataSet As IGeoDataset
Set pGeoDataSet = pInputFeatureClass

Dim pSR As ISpatialReference
Set pSR = pGeoDataSet.SpatialReference
```

Listing 1: Deriving the spatial reference

The multipatch is a three-dimensional feature, so it is essential that the Z Domain in the SpatialReference of the output feature class is set. The GeometryType property of the Geometry definition in the Shape field of the feature class must be set to

esriGeometryMultiPatch.

Once the output feature class is created, it can then be populated with the three-dimensional features. As outlined above, a multipatch feature consists of a number of triangle strips or triangle fans. In ArcObjects, these must be created initially and populated with the input points in the correct order, then added to the multipatch feature. The order in which the points are added depends on whether the triangle strip or triangle fan is used. A triangle strip assumes that every point added creates a triangle in conjunction with the previous two points. For most applications, the triangle fan results in the simplest implementation.

Initially an IMultipatch feature is declared in ArcObjects, along with an IGeometryCollection and an IPointCollection.

```
Dim pMultiPatch As IMultiPatch
Dim pGCol As IGeometryCollection
Dim pStrip As IPointCollection

Set pMultiPatch = New MultiPatch
Set pGCol = pMultiPatch
Set pStrip = New TriangleStrip
```

Listing 2: Creating a triangle strip multipatch

The Geometry Collection object is used as an interface to the multipatch feature, since it is essentially a collection of triangle strip and triangle fan geometries. Each triangle strip or fan is essentially a collection of Point features, hence the need for the IPointCollection object.

Each point in turn is then added to the Point Collection object in the correct order, until the triangle strip or fan is completed. It is essential for the points that make up a triangle in the object to be ordered clockwise for ArcObjects to determine which side of each triangle is the outside and compute normals. *[The normal vector of a surface is one that is perpendicular to the plane of that surface. This defines the orientation of the surface in space and relation to light sources. These are required for correct three-dimensional visualizations later.]*

```
pStrip.AddPoint pTempPoint1
pStrip.AddPoint pTempPoint2
pStrip.AddPoint pTempPoint3

pGCol.AddGeometry pStrip
```

Listing 3: Adding the points that make up the triangle strip

Once all points are added to the triangle strip or fan, this geometry is then added to the Geometry Collection object (and hence the Multipatch feature). The final step ensures that the Multipatch is Z Aware (as it is a three-dimensional object) and adds the feature to the

Continued on page 18

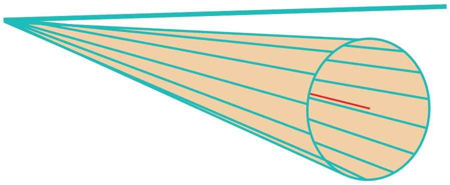
Visualizing Integrated Three-Dimensional Petroleum Datasets

Continued from page 17

output feature class through an Insert Feature Buffer as normal.

The Pipeline Object

The pipeline object was created from a series of points representing the centerline of the pipe, extracted from swathe bathymetry data from an inspection survey. These were read into ArcGIS through ArcCatalog and a 3D Point feature class created from the x,y,z coordinates. In ArcObjects, an IVector3D object was created for each pair of points defining the pipeline, and a second vector was created perpendicular to this with a magnitude equal to the required diameter of the pipe. The second vector was then rotated around the first, creating a new IPoint object a distance equal to the diameter away from the centerline of the pipe. This created a new ring of points at each vertex along the pipe that could then be added in the correct order to a new multipatch feature to create a tube representation.

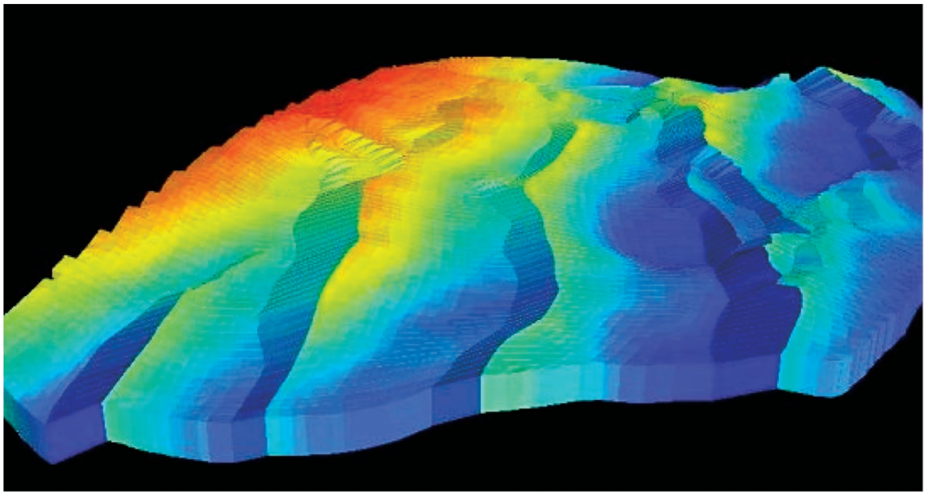


Multipatch pipeline created from points defining centerline

Reservoir Object

While the creation of a pipeline object was relatively straightforward, an oil or gas reservoir feature has greater complexity. Schlumberger's geological modeling software Petrel was chosen because it is widely used and Windows based. Petrel uses "faulted grids" in its reservoir models, which allow the user to define irregular variations throughout a solid structure. These grids are composed of three-dimensional cells of irregular shape and size, the number of rows, columns, and layers throughout an object being constant rather than cell size.

A Petrel model was exported in Eclipse ASCII grid format, which defines the two opposite corner points of each cell. These, in conjunction with the corner points of neighboring cells, can be used to build up the irregular structure of the faulted grids. The corner points of each cell were loaded into ArcGIS as x,y,z points as before. A number of nested loops were then written in ArcObjects to parse each cell and the corresponding neighbors in the correct order, adding each point to a triangle strip as before. A multipatch feature was created for each cell, allowing each an individual record in the geodatabase table and therefore allowing the assignment of attributes on a cell-by-cell basis.

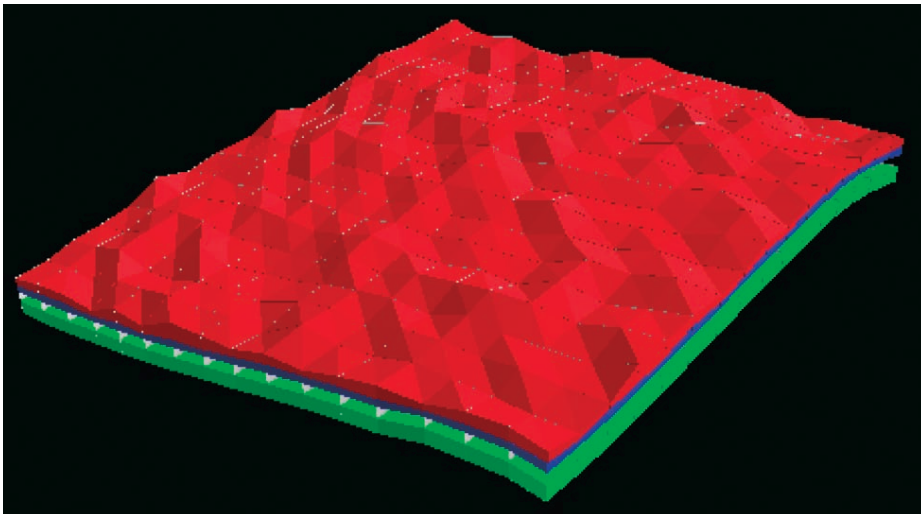


Faulted grid stored as multipatch in an ESRI geodatabase

Once multipatch features have been created within the ESRI geodatabase, their behavior in ArcMap or ArcScene is essentially identical to any other feature type. The symbology of a multipatch feature may be set through the normal Layer Properties dialog box, allowing the user to alter the display of the multipatch dependent on any of the attributes in the geodatabase table. While in most geological modeling software, each cell is assigned only one property (or attribute), ArcGIS allows the storage of multiple attributes about each cell and the interactive display of these properties through symbology.

Data Volumes

A geological application was deliberately chosen as a case study as it offers the greatest challenge to GIS developers, not only because of the complexity of the objects to be modeled but also due to the vast quantities of data to be stored. Each cell in the faulted grid model above is represented by 12 triangles in a multipatch feature. A grid with a reasonably coarse resolution may still consist of 50 x 50 x 10 cells, which would require 300,000 triangles to represent it in this manner.



Cells displayed by geological layer

Issues and Limitations

It can be seen from previous sections that the multipatch feature allows for the storage of complex three-dimensional features in an ESRI geodatabase. While this is the first step toward true three-dimensional objects in a GIS, there are still a number of limitations with this approach. These limitations are outlined in the following sections

It is instantly evident that this is an impractical means of storing such large quantities of data. Similarly, 1.5 kilometers of pipeline stored as a simple line feature is barely 1 MB of data, while the same pipeline stored as a multipatch feature (even with a low level of detail) can amount to more than 100 MB.

There are numerous other data types in the oil and gas industry, such as seismic data, that require huge volumes of storage space. Data storage in most geological packages has remained relatively simple in a bid to improve performance (often using flat ASCII files for storage), while for GIS this has been less of a concern.

One solution to this problem would be to mimic the geological modeling software and limit storage of such data to simple point features, creating the complex multipatch representations on the fly when complex visualizations or analysis is needed. This would require more work at runtime but would reduce the storage cost. It is obvious that the storage of all pipeline features as true three-dimensional objects is impractical, whereas storage of a line feature representing a pipe, with a numerical attribute representing the pipe diameter, would allow for on-the-fly creation of the required section of pipeline at the time of visualization or analysis.

Three-Dimensional Symbology

The introduction of three-dimensional symbology at ArcGIS 9 has rendered the creation of multipatch representations of features for visualizations trivial. For example, voxel data (uniform three-dimensional arrays of x,y,z points with attributes) can be stored as a grid of points and symbolized using three-dimensional symbology, thus vastly saving storage costs. To carry out three-dimensional spatial analysis, however, it is essential for the three-dimensional geometry of objects to be existent and accessible rather than simply visualized.

One area where three-dimensional symbology is particularly powerful is in the creation of standardized libraries of three-dimensional objects, which can be included in any visualization. Complex three-dimensional objects, such as oil platforms, refineries, well heads, or pipe components, can now be created from this common library and simply stored as point features with the necessary attributes to assign location and orientation to the three-dimensional object.

Representing Variation

One interesting problem in the geosciences is the representation of the spatial variation of attributes. It was noted above that in some geological modeling software, variation is represented by dividing an object into cells. In GIS, features are traditionally represented by either vector (discrete) or raster (continuous) data. In some cases this is insufficient. For example, a reservoir is a discrete entity with boundaries, but the attributes of this object vary spatially throughout. Therefore, it is not sufficient to store a single object with one attribute for permeability, as this attribute itself varies depending on the location within the object.

In one-dimensional objects (i.e., line features), it is currently possible to represent this. The use of dynamic segmentation allows the spatial variation of attributes along the length of a line. Through this technique, it is possible to illustrate how the characteristics of a linear feature vary through space. For example, along a well feature it is possible to illustrate the variation of a lithological layer or even returns from a well log. This can be achieved by referencing a table of data defining the attributes at given distances along a line to the measures along the line geometry itself.

In the current GIS mindset, however, it is not possible to account for both raster and vector representations of a two-dimensional or three-dimensional object. While variation in two-dimensional space can be represented by raster features and in three-dimensional space by voxels, it is not possible to combine both of the representations into one feature in a database. The need for this hybrid representation is clearly one area of future work in three-dimensional object modeling.

Conclusion

This article has discussed some of the many issues that must be resolved to include true three-dimensional objects in an ESRI geodatabase. It has outlined the data modeling considerations and basics of three-dimensional object modeling, before describing the methodology for implementing such a model in the geodatabase using multipatch features. It can be seen that there is much work to be done to provide comprehensive three-dimensional functionality in commercial GIS software. In addition to data storage, advances much be made in visualizations and spatial analysis in three dimensions before any GIS can truly call itself 3D. For more information, contact Alistair Ford at A.C.Ford@ncl.ac.uk.

Acknowledgments

The author thanks Professor David Parker, Mr. Philip James, and Dr. David Fairbairn at the University of Newcastle upon Tyne's School of Civil Engineering and Geosciences and Dr. Thierry Gregorius and other staff of Shell International Exploration and Production for their assistance and support. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Petrel is a trademark of Schlumberger Information Services.

About the Author

Alistair Ford, a research associate at Newcastle University in the United Kingdom for the past two years, has been working on a variety of GIS projects in the School of Civil Engineering and Geosciences. He received a BSc with honors in geographical information science in 2001 and studied for a doctorate, sponsored by Shell, in three-dimensional GIS for data integration in the oil and gas industry. He is currently completing this study while also working on a number of research projects including climate modeling and land-use simulation. He is also part of the Tyndall Centre for Climate Change Research where he is investigating land-use change in London to 2100 and the effects of this on water resources, flooding, and emissions.

References

Abel, D. J., P. J. Kilby, and J. R. Davis (1994), "The Systems Integration Problem," *International Journal of Geographical Information Systems,* 8 (1), 1–12.

Breunig, M. (1998), "An Approach to the Integration of Spatial Data and Systems for a 3D Geo-information System," *Computers and Geosciences,* 25 (1), 39–48.

Carlson, E. (1987), "Three Dimensional Conceptual Modeling of Subsurface Structures," *Technical Papers of ASPRS/ACSM Annual Convention,* Vol. 4 (Cartography), Baltimore, Maryland, 188–200.

de la Losa, A., and B. Crevelle (1999), "3D Topological Modeling and Visualizations for 3D GIS," *Computers and Graphics,* 23(4), 469–478.

Molenaar, M. (1990), "A Formal Data Structure for 3D Vector Maps." *Proceedings of EGIS'90,* Vol. 2, Amsterdam, The Netherlands, 770–781.

Sloter, J., and P. a. v. Oosterom (2002), "Incorporating 3D Geo-Objects into a 2D Geo-DBMS," *ACSM-ASPRS 2002 Annual Conference Proceedings,* Washington, D.C.

Abdul-Rahman, A., S. Zlatanova, and W. Shi (2002), "Topology for 3D Spatial Objects," *Proceedings of International Symposium and Exhibition on Geoinformation 2002,* Kuala Lumpur.