Data Model Overview

A data model is a set of constructs for representing objects and processes in the digital environment of the computer.

Decisions about the type of data model to be adopted are vital to the success of a GI project.

Modern GI systems are able to incorporate multiple data models.
A data model provides:

- developers with the means to represent an application domain in terms that may be translated into a design and implemented
- users with a flexible system, understandable independent of specific items of data or details of the particular application

Important (dangerous?) decisions:

- must define the main types of “things” to be represented in the GI database
- must understand how the “things” interact
Digital Representation
## GI Data Models

<table>
<thead>
<tr>
<th>Data model</th>
<th>Example application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer-aided design (CAD)</td>
<td>Automating engineering design and drafting</td>
</tr>
<tr>
<td>Graphical (non-topological)</td>
<td>Simple mapping and graphic arts</td>
</tr>
<tr>
<td>Image</td>
<td>Image processing and simple grid analysis</td>
</tr>
<tr>
<td>Raster/grid</td>
<td>Spatial analysis and modeling, especially in environmental and natural resources applications</td>
</tr>
<tr>
<td>Vector/georelational topological</td>
<td>Many operations on geometric features in cartography, socioeconomic, and resource analysis and modeling</td>
</tr>
<tr>
<td>Network</td>
<td>Network analysis in transportation and utilities</td>
</tr>
<tr>
<td>Triangulated irregular network (TIN)</td>
<td>Surface/terrain visualization, analysis, and modeling</td>
</tr>
<tr>
<td>Object</td>
<td>Many operations on all types of entities (raster/vector/TIN, etc.) in all types of applications</td>
</tr>
</tbody>
</table>
Computer Aided Design

CAD system: real-world entities are represented symbolically as simple point, polyline, and polygon vectors.

Three significant problems with CAD models:
- Use local drawing coordinates.
- Individual objects do not have unique identifiers.
- Focused on graphical representation of objects, and cannot store relationships.
Graphical and Image Models

Graphical data models:
- store entities as points, polylines, and polygons
- annotation used for placenames
- adobe illustrator

Image data models:
- store scanned aerial photos and digital satellite images as rasters or grids
- adobe photoshop
Raster Data Model

Uses an array of cells or pixels to conveniently represent continuous spatial phenomena

Cells can hold any attribute values based on one of several encoding schemes including categories, integer, and floating-point numbers

In some systems, multiple attributes can be stored, multiple bands

Often an efficient storage format with many techniques for compressing
# Raster Data Types

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>point ID</td>
<td>alpha-numeric ID of closest point</td>
<td>nearest hospital</td>
</tr>
<tr>
<td>line ID</td>
<td>alpha-numeric ID of closest line</td>
<td>nearest road</td>
</tr>
<tr>
<td>contiguous region ID</td>
<td>alpha-numeric ID for dominant region</td>
<td>state</td>
</tr>
<tr>
<td>class code</td>
<td>alpha-numeric code for general class</td>
<td>vegetation type</td>
</tr>
<tr>
<td>table ID</td>
<td>numeric position in a table</td>
<td>table row number</td>
</tr>
<tr>
<td>physical analog</td>
<td>numeric value representing surface value</td>
<td>elevation</td>
</tr>
<tr>
<td>statistical value</td>
<td>numeric value from a statistical function</td>
<td>population density</td>
</tr>
</tbody>
</table>
ESRI’s ASCII file

A text file with two parts:
  header
  data

There are other common file formats:
  GeoTIFF
  HDF
  image

See fake_raster.txt
How to get a cell’s location?

- Coordinates of lower-left cell
- Cell dimension
Dealing with Large Data

Raster data sets can be very large, with a lot of repetitive information.

There are many compression techniques to reduce storage needs:
- run-length encoding: one row at a time
- block encoding: block by block
- wavelet
Dealing with Large Data

Run-length encoding:

- A string of 100 5’s will be encoded as 100:5
- A string of 20 5’s and 80 6’s becomes 20:5, 80:6

<table>
<thead>
<tr>
<th>Raster</th>
<th>Run-length codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>9 9 6 6 6 6 6 6 7</td>
<td>2:9, 5:6, 1:7</td>
</tr>
<tr>
<td>6 6 6 6 6 6 6 6</td>
<td>8:6</td>
</tr>
<tr>
<td>9 9 6 6 6 6 6 7 7</td>
<td>2:9, 4:6, 2:7</td>
</tr>
<tr>
<td>9 8 9 6 6 7 7 5</td>
<td>1:9, 1:8, 1:9, 2:6, 2:7, 1:5</td>
</tr>
</tbody>
</table>
Block encoding, aka Region Quadtrees

The *root* of a tree represents the entire region

If the entire region is not homogeneous, then it is split into four equal quadrangles

We keep splitting each quadrangle into sub-quadrangles until it is homogeneous

When a quadrangle needs to split, we use a non-leaf node to represent it

otherwise, we use a leaf node to represent a homogeneous region
Data Models: Not just Vector & Raster
Why Quadtree?

How do we use a quadtree?
search for the value of at a certain location given the current location in the tree

Advantages of a quadtree
data compression (nice)
optimized (i.e. very fast) queries (very nice!)

But there are potential disadvantages
complexity of the original raster data impacts the depth of a quadtree
construction of quadtree is time consuming
Open Franklin County.mxd

Make sure that the Shapefiles data frame is active

Run:

`rasterWork.R` ⇒ pulls data from a raster
Raster Data Worth Looking At

NBA “heat maps” show where players spend more time (red) and less time (blue)


\(^1\)Cervone, Bornn, and Goldsberry
Vector Data Model

Each object in the real world is first classified into a geometric type in the 2-D case: point, line, or area.

Very convenient for modeling discrete objects with discrete elements:

- points are recoded as single coordinate pairs
- lines as a series of ordered coordinate pairs
- areas as one or more line segments that close to form a polygon

In some systems, curves can be defined by a mathematical function.
What’s a Vector?

From the TA: a quantity having direction as well as magnitude, especially as determining the position of one point in space relative to another
Simple features in the Vector Model

Geographic entities encoded using the vector data model are usually called \textit{features}.

Features of the same geometric type are stored in a geographic database as a feature class, or when speaking about physical representation the term feature table is preferred.

Simple feature datasets are sometimes called \textit{SPAGHETTI} because lines and polygons can overlap and there are no relationships between any of the objects.

Simple features lack more advanced data structure characteristics, such as topology.
Spaghetti Vector Models

We call this a spaghetti data model because each object is modeled individually.

But this may be inefficient.

Is polygon 1 adjacent with polygon 2?
Simple Vector Data Storage: Shapefiles

Developed by ESRI for use with the dBASE III database management system

- simple, nontopological files
- stores the geometric location and attribute information of geographic features

Each shapefile only represents point, line, or polygon feature sets

Attribute tables:
- supported data types are limited to floating point, integer, date, and text
- field names within the attribute table are limited to ten characters

Supported by almost all commercial and open-source GIS software
### Simple Vector Data Storage: Shapefiles

<table>
<thead>
<tr>
<th>File Extension</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHP*</td>
<td>Feature geometry</td>
</tr>
<tr>
<td>SHX*</td>
<td>Index format for the feature geometry</td>
</tr>
<tr>
<td>DBF*</td>
<td>Feature attribute information in dBASE IV format</td>
</tr>
<tr>
<td>PRJ</td>
<td>Projection information</td>
</tr>
<tr>
<td>SBN and SBX</td>
<td>Spatial index of the features</td>
</tr>
<tr>
<td>FBN and FBX</td>
<td>Read-only spatial index of the features</td>
</tr>
<tr>
<td>AIN and ALH</td>
<td>Attribute information for active fields in the table</td>
</tr>
<tr>
<td>IXS</td>
<td>Geocoding index for read-write shapefiles</td>
</tr>
<tr>
<td>MXS</td>
<td>Geocoding index for read-write shapefiles with ODB format</td>
</tr>
<tr>
<td>ATX</td>
<td>Attribute index used in ArcGIS 8 and later</td>
</tr>
<tr>
<td>SHP.XML</td>
<td>Metadata in XML format</td>
</tr>
<tr>
<td>CPG</td>
<td>Code page specifications for identifying character encoding</td>
</tr>
</tbody>
</table>

* Indicates mandatory files
Open Franklin County.mxd

Make sure that the Shapefiles data frame is active

Run:

pointVertices.py ⇒ pulls coordinates of point features
polylineVertices.py ⇒ pulls vertices of line features
polygonVertices.py ⇒ pulls vertices of area features
Topological Vector Models

Simple features structured using topological rules

Topology is the mathematics and science of geometrical relationships between objects:

adjacency, containment, intersection, direction

Topological relationships are the qualitative properties of geographic objects that remain constant even when the geographic space of objects is distorted

In a topologically structured polygon data layer each polygon is defined as a collection of polylines that in turn are made up of an ordered list of coordinates (vertices)
Topological Vector Models

Data validation topology tests include network connectivity, line intersection, overlap, duplicate lines.

Topological structuring of layers forces all line ends that are within a user-defined distance to be snapped together so that they are given exactly the same coordinate value.

- A node is placed wherever the ends of lines meet or cross.
- Avoids the potential problems of gaps (slivers) or overlaps.

Downside is that drawing polygons from multiple polylines is time intensive.
Topologically Structured Polygon Layer

Area-polyline topology

Area-polyline list

<table>
<thead>
<tr>
<th>POLY</th>
<th>POLYLINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4, 6, 7, 10, 0, 8</td>
</tr>
<tr>
<td>3</td>
<td>3, 10, 9</td>
</tr>
<tr>
<td>4</td>
<td>7, 5, 2, 9</td>
</tr>
<tr>
<td>5</td>
<td>1, 5, 6</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
</tr>
</tbody>
</table>

Polyline coordinate list

<table>
<thead>
<tr>
<th>POLYLINE (x,y) coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
</tbody>
</table>
Contiguity Stored in Topologically Structured Data
Using a Topologic Vector Model

The topological data structure is efficient in data storage.

Helps us find useful information quickly:
- are polylines 3 and 7 connected?
- are polygons A and B connected?
- shortest path between nodes 17 and 8?

Need non-topological information such as distance between nodes, but topology is fundamental.
Validating Topology

Node chaining:
  focusing on a polygon, counting the surrounding nodes clockwise:
  A: 4, 23, 28, 17, 4

Block chaining:
  focusing on a node, counting the surrounding polygons clockwise
  28: A, C, D, A
What’s So Good About Topology?

a) spaghetti

b) topological

c) topological - warped
In the georelational model:

- the feature geometries and associated topological information are stored in “regular computer files”
- the associated attribute information is held in relational database management system (RDBMS) tables

Geometry and topology were not placed in RDBMS because until relatively recently RDBMS were unable to store and retrieved geographic data efficiently.

Exact georeferences are held in the “tics”
Georelational Polygon Dataset

Soils layer

+1 + 5 label point
+2
+3
polyline
+6
+7
node

tic

Soils attributes

<table>
<thead>
<tr>
<th>ID</th>
<th>Soil</th>
<th>Class</th>
<th>Suitability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A3</td>
<td>113</td>
<td>HIGH</td>
</tr>
<tr>
<td>2</td>
<td>C6</td>
<td>95</td>
<td>LOW</td>
</tr>
<tr>
<td>3</td>
<td>B7</td>
<td>212</td>
<td>MODERATE</td>
</tr>
<tr>
<td>4</td>
<td>B13</td>
<td>201</td>
<td>MODERATE</td>
</tr>
<tr>
<td>5</td>
<td>Z22</td>
<td>86</td>
<td>LOW</td>
</tr>
<tr>
<td>6</td>
<td>A6</td>
<td>77</td>
<td>HIGH</td>
</tr>
<tr>
<td>7</td>
<td>A1</td>
<td>117</td>
<td>LOW</td>
</tr>
</tbody>
</table>
Topological Vector Models

Simple features + topological rules

Useful because of role in:
  - data validation
  - modeling the integrated behavior of different feature types
  - editing productivity
  - optimizing queries!
A coverage is a georelational data model that stores vector data containing both:

- spatial (location) data
- attribute (descriptive) data

Coverages use a set of feature classes to represent geographic features

Each feature class stores a set of points, lines (arcs), polygons, or annotation (text)

A coverage is stored as a directory within which each feature class is stored as a set of files
In this example, you can see that the streams coverage is a line coverage containing:

- an arc (line) file
- annotation for the line
- a tic file
ArcMap Break Again

Open Franklin County.mxd

Check out the County Coverage and Tract Coverage data frames
Vector Data Worth Looking At

La Liga (2012-2013) “vector maps” of players in game directionality

Spaghetti or Topological?

---

2Gyarmati and Hefeeda
Is Vector “Corrector”? 

Spatial distribution of rebounds from shots missed from the left-corner-three area (NBA 2013-2014)

Which is vector, which is raster? Is the vector version more accurate in this case?³

³Maheswaran, Chang, Su, Kwok, Levy, Wexler, and Hollingsworth
Network Data Model

A vector data format

Network topological relationships define how lines connect with each other at nodes and define rules about how flows can move through the network.

In linear referencing systems, the locations of geographic entities are stored as distances along a network from a point of origin.

Dynamic segmentation is a special case of linear referencing in which data values are added dynamically to the route each time the user queries the database.
Street Network Example

Data Models: Not just Vector & Raster
TIN Model

Triangulated irregular networks

A TIN is a topological data structure that manages information about the nodes comprising each triangle and the neighbors of each triangle.

TIN surfaces often created by performing Delaunay triangulation.
A TIN is a topologic data structure that manages information about the nodes that comprise each triangle and the neighbors to each triangle.

Triangles always have three nodes and usually have three neighboring triangles. Triangles on the periphery of the TIN can have one or two neighbors.

<table>
<thead>
<tr>
<th>Triangle</th>
<th>Node list</th>
<th>Neighbors</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1, 2, 3</td>
<td>-, B, D</td>
</tr>
<tr>
<td>B</td>
<td>2, 4, 3</td>
<td>-, C, A</td>
</tr>
<tr>
<td>C</td>
<td>4, 8, 3</td>
<td>-, G, B</td>
</tr>
<tr>
<td>D</td>
<td>1, 3, 5</td>
<td>A, F, E</td>
</tr>
<tr>
<td>E</td>
<td>1, 5, 6</td>
<td>D, H, -</td>
</tr>
<tr>
<td>F</td>
<td>3, 7, 5</td>
<td>G, H, D</td>
</tr>
<tr>
<td>G</td>
<td>3, 8, 7</td>
<td>C, -, F</td>
</tr>
<tr>
<td>H</td>
<td>5, 7, 6</td>
<td>F, -, E</td>
</tr>
</tbody>
</table>
Voronoi to Delaunay

Voronoi polygon zonation divides space so that all points fall within only one zone ⇒ the zone it is closest to

Delaunay triangulation zonation divides space into triangular regions

Very nice mathematical properties!
TIN Representation of Topography
TIN Model

Advantages:
- density of sampled points can be adjusted to reflect relief
- incorporate the original sample points
- easy to calculate elevation, slope, aspect, and line-of-sight

Limitations:
- susceptible to extreme high and low values since there is no smoothing of the original data
- unable to deal with discontinuity of slope across triangle boundaries
- difficult to calculate optimum routes
Object Data Model

An object is a self-contained package of information describing the characteristics and capabilities of an entity under study:

- each geographic object is an integrated package of geometry, properties, and methods
  ⇒ geometry is treated like any other attribute of the object

Geographic objects of the same type are grouped together as object classes:

- a class can be thought of as a template for objects.
- individual objects in the class are instances

An interaction between two objects is called a relationship.
Object Model: Waterlines

- Main
- Pump
- Fitting
- Meter
- Lateral
- Valve
- Hydrant
No step in data modeling is more important than understanding the purpose of the data modeling exercise, gained by collecting user requirements from the main users.

Once an implementation-independent logical model has been created (not just raster or vector), this model can be turned into a system-dependent physical model (data on disk).
Other Data Formats: Layer Files

A layer file (.lyr) is a set of rules for displaying geospatial data:
- symbol assignments
- classifications
- labeling rules

Layer files are not new datasets, but rather a specification for displaying existing datasets:
- we’ve already seen that layer files are “sourced” to a specific dataset ⇒ like a shapefile

Layer files cannot be edited, though you can select and export data into a new shapefile.
I prefer scarlet for points and gray for polygons

We can get very precise in our color palette decisions:

http://brand.osu.edu/color/

We may want to use this custom symbology for later applications!
Data Models: Not just Vector & Raster

Jake K. Carr
Data Models: Not just Vector & Raster
Other Geospatial Data Formats

ArcInfo Interchange Files (.e00)

an ESRI proprietary data format allowing portability of
geospatial data from ArcInfo (an older component of ArcGIS)
and other GIS software packages
not especially common since, but still found in some
geospatial data depots

Easy to add to ArcMap, convert to shapefile/feature class

ArcToolbox ⇒ Conversion Tools ⇒ To Coverage ⇒ Import
from E00
Other Geospatial Data Formats

Google Earth Keyhole Markup Language (\textit{.kml})

used to display geographic data in an Earth browser such as Google Earth, Google Maps, and Google Maps for mobile
these tools provide excellent geospatial data visualization and very simple GIS capabilities at little to no cost
increasingly common, since the free availability of Google Earth is “democratizing” geospatial data usage

Easy to add to ArcMap, convert to shapefile/feature class

\texttt{ArcToolbox $\Rightarrow$ Conversion Tools $\Rightarrow$ From KML $\Rightarrow$ KML to Layer}
For Next Week!

Read Chapter 9 from Longley et al.\textsuperscript{4}

Finish Lab 4

Bring Lab book with you on Tuesday!

\textsuperscript{4}Lecture slides adapted from Longley et al.