Representing the real world in a GIS: how geographic information is stored in the computer
Representing the real world in a GIS

- the world is infinitely complex
- the contents of a spatial database represent a limited view of reality
  -> the spatial database is a model of reality
- the user sees the real world through the medium of the database
Representing the real world

- a database may include
  - digital versions of real objects
e.g., houses, roads, forests
  - digital versions of fictitious (i.e., invented) objects
e.g., political boundaries
Representing the real world

- computers are good at storing discrete spatial data, but bad at storing continuous data
Representing the real world

• some features are discrete, clearly defined entities (e.g., houses, districts)
  -> discrete representation is no problem

• other features exist everywhere and vary continuously (e.g., temperature)
  -> variation needs to be approximated using discrete representations
sometimes, the distinction between discrete and continuous is not very clear
Objects versus Fields

- Object view
  “empty space littered with objects”
  (points, lines or areas)

- Field view
  value is defined for every location
Example of a field

• digital elevation models (DEM)

Elevation in Nepal
Objects

Points

Lines

Polygons
Data model implementation: the vector data model
The data model

• rules to convert real geographic variation into discrete representations
GIS data models

- two major types:
  - raster data model
  - vector data model

- raster data model will be discussed later
The vector data model

- real world objects are represented as points, lines and areas
- points identify locations
- lines connect points
- areas (polygons) consist of connected line segments
The vector data model

- Points
- Lines
- Areas

Node
Vertex
The vector data model

- objects are defined by their x/y coordinates in the planar (Cartesian) coordinate system
- precision of coordinates virtually infinite (only machine-dependent)
- but: accuracy most often limited!
Precision versus Accuracy

• **precision** is the ability to distinguish between small quantities or distances in measurement

• **accuracy** is freedom from error
“Spaghetti” data model

- **point** is recorded as x,y coordinate pair
- **line** is a series of x,y coordinates
- **area** is a series of x,y coordinates, with the first and last coordinate being identical (e.g., “closed-loop polygons”)
<table>
<thead>
<tr>
<th>Area</th>
<th>Coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>(1,4), (1,6), (6,6), (6,4), (4,4), (1,4)</td>
</tr>
<tr>
<td>B</td>
<td>(1,4), (4,4), (4,1), (1,1), (1,4)</td>
</tr>
<tr>
<td>C</td>
<td>(4,4), (6,4), (6,1), (4,1), (4,4)</td>
</tr>
</tbody>
</table>

“Spaghetti” Data Model

Points and lines would be encoded in a similar way; **note**: there is no relationship between points, lines and areas.
Topological data model

- records x/y coordinates of spatial features and encodes spatial relationships
- also called “arc-node” data model
- arc = line
- node = end-point of a line, or a point where two or more lines connect
### Topological Data Structure

<table>
<thead>
<tr>
<th>Node</th>
<th>X</th>
<th>Y</th>
<th>Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1</td>
<td>4</td>
<td>1,2,4</td>
</tr>
<tr>
<td>II</td>
<td>4</td>
<td>4</td>
<td>4,5,6</td>
</tr>
<tr>
<td>III</td>
<td>6</td>
<td>4</td>
<td>1,3,5</td>
</tr>
<tr>
<td>IV</td>
<td>4</td>
<td>1</td>
<td>2,3,6</td>
</tr>
</tbody>
</table>

**Poly Lines**
- A: 1, 4, 5
- B: 2, 4, 6
- C: 3, 5, 6

**Lines**

<table>
<thead>
<tr>
<th>Line</th>
<th>From</th>
<th>To</th>
<th>Left</th>
<th>Right</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I</td>
<td>III</td>
<td>O</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>I</td>
<td>IV</td>
<td>B</td>
<td>O</td>
</tr>
<tr>
<td>3</td>
<td>III</td>
<td>IV</td>
<td>O</td>
<td>C</td>
</tr>
<tr>
<td>4</td>
<td>I</td>
<td>II</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>5</td>
<td>II</td>
<td>III</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>6</td>
<td>II</td>
<td>IV</td>
<td>C</td>
<td>B</td>
</tr>
</tbody>
</table>

O = “outside” polygon
Topological data model can quickly answer these questions:

• which roads connect to the central square?
• which roads do I take to get from here to the hospital?
• what are the fertility rates in the neighboring districts?
Storing attribute data

- attribute data are stored separately from the coordinate data

- feature identifier points to an attribute table:
  - **point** attribute table
  - **line** or **arc** attribute table
  - **polygon** attribute table
Storing attribute data

Similarly we can define point or line attribute tables if the spatial features are, for example, villages and roads.
<table>
<thead>
<tr>
<th>Id</th>
<th>Province</th>
<th>District</th>
<th>P_Pop</th>
<th>P_TFR</th>
<th>D_Pop</th>
<th>D_TFR</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>Merida</td>
<td>Palma</td>
<td>214084</td>
<td>3.2</td>
<td>89763</td>
<td>3.4</td>
</tr>
<tr>
<td>102</td>
<td>Merida</td>
<td>S. Maria</td>
<td>214084</td>
<td>3.2</td>
<td>45938</td>
<td>2.9</td>
</tr>
<tr>
<td>103</td>
<td>Merida</td>
<td>Veralo</td>
<td>214084</td>
<td>3.2</td>
<td>78383</td>
<td>3.2</td>
</tr>
<tr>
<td>104</td>
<td>La Paz</td>
<td>Bolo</td>
<td>397881</td>
<td>3.7</td>
<td>98302</td>
<td>3.9</td>
</tr>
<tr>
<td>105</td>
<td>La Paz</td>
<td>Jose</td>
<td>397881</td>
<td>3.7</td>
<td>67352</td>
<td>4.2</td>
</tr>
<tr>
<td>106</td>
<td>La Paz</td>
<td>Malabo</td>
<td>397881</td>
<td>3.7</td>
<td>102839</td>
<td>3.7</td>
</tr>
<tr>
<td>107</td>
<td>La Paz</td>
<td>Chilabo</td>
<td>397881</td>
<td>3.7</td>
<td>129388</td>
<td>2.8</td>
</tr>
</tbody>
</table>

Storing the province and district data in the same table is inefficient, because province data need to be repeated for each district.
Storing attribute data

- instead we can produce a more efficient database that does not include as much redundancy
- relational database
- process to separate variables into several files is called normalization
<table>
<thead>
<tr>
<th>Id</th>
<th>District</th>
<th>D_Pop</th>
<th>D_TFR</th>
<th>Province</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>Palma</td>
<td>89763</td>
<td>3.4</td>
<td>Merida</td>
</tr>
<tr>
<td>102</td>
<td>S. Maria</td>
<td>45938</td>
<td>2.9</td>
<td>Merida</td>
</tr>
<tr>
<td>103</td>
<td>Veralo</td>
<td>78383</td>
<td>3.2</td>
<td>Merida</td>
</tr>
<tr>
<td>104</td>
<td>Bolo</td>
<td>98302</td>
<td>3.9</td>
<td>La Paz</td>
</tr>
<tr>
<td>105</td>
<td>Jose</td>
<td>67352</td>
<td>4.2</td>
<td>La Paz</td>
</tr>
<tr>
<td>106</td>
<td>Malabo</td>
<td>102839</td>
<td>3.7</td>
<td>La Paz</td>
</tr>
<tr>
<td>107</td>
<td>Chilabo</td>
<td>129388</td>
<td>2.8</td>
<td>La Paz</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P_Pop</th>
<th>P_TFR</th>
<th>Province</th>
</tr>
</thead>
<tbody>
<tr>
<td>397881</td>
<td>3.7</td>
<td>La Paz</td>
</tr>
<tr>
<td>214084</td>
<td>3.2</td>
<td>Merida</td>
</tr>
</tbody>
</table>

a relational database design provides better storage efficiency
Storing attribute data

• good organization of the attribute data is very important
• in socioeconomic GIS applications, the attribute data component is often much larger than the database component; e.g., few provinces, but hundreds of variables