Making features “smart”

GEOG 419/519: Advanced GIS
Dr. Lembo
Qualities of features (p. 76)

• Features have shapes
  – Geometry feature class
    • Points and multipoints
    • Polylines (line segments that may or may not be connected)
    • Polygons (set of either disjoint or embedded rings)
      – Ring: set of connected, closed, nonintersecting line segments
    • Line segments can be straight, circular arcs, Bézier curves, and elliptical arcs
Qualities of features II (p. 76)

- **Features have spatial reference**
  - Specifies how the x,y coordinates of a set of features are mapped onto the earth’s surface

- **Features have attributes**
  - Fields in a feature class table
  - Standard or custom properties of features
  - Numeric, textual, descriptive

- **Features have subtypes**
  - Feature class is homogenous set of features
    - May be considerable variation among features
  - Buildings → Residential, Commercial, Industrial
  - Subtypes give more control over qualities of features
    - No longer all or nothing…
Qualities of features III (p. 77)

• Features have relationships
  – Among geographic features
  – Between geographic features and nonspatial objects

• Features can be constrained
  – Attribute domain: numeric range or list of valid values
    • All or nothing…

• Features have rule validation
  – Pipe connections

• Features have topology
  – Precise spatial relationships
  – Planar topology
  – Geometric network

• Features have complex behavior
  – Custom editing interaction, intrinsic analytical capabilities, sophisticated cartographic rendering
Steps to making features smart (p. 78)

- Progressively adding intelligence
- Select feature type and topology
  - Inventory all of the types of objects you need to model
  - Establish feature datasets to group feature classes that are bound by spatial reference, topology, and thematic content (Chambers article)
    - Nonspatial: create object classes
    - Spatial: create simple feature classes (point, line polygon)
    - Topological: create a graph with topological feature classes in a common feature dataset
Steps to making features smart II

• Set attribution and subtypes (p. 89)
  – Subtype – special attribute
    • Major groupings of objects
    • Can express diversity among similar objects/features without creating lots of classes (the more classes, the slower the geodatabase)
    • Improve data integrity with attribute domains, default values, connectivity rules, relationship rules

• Define attribute domains and validation rules

• Establish object relationships
  – Relationship classes, relationship rules

• Create custom objects
  – VisualBasic programming
Geodb design considerations
(Chambers article)

• Keep features/objects in same geodb if (p. 195):
  – Objects/features have relationships
  – Features that have topological associations
  – In order to concurrently edit features
    • Can only edit one geodb at a time

• Separate features/objects in separate geodb if:
  – Different departments have responsibility for different datasets
  – Use of different commercial RDBMS
  – Personal geodb have practical size limits
• Classes (object, feature, relationship) can either stand alone or reside in a feature dataset

• Group classes in a feature dataset if:
  – If feature classes are topologically related
  – If you wish to enforce a common spatial reference for a set of feature classes
  – Logical organization
Geoddb design considerations

III

- Should a group of related features be separate feature classes or subtypes of the same feature class?
- The more feature classes, the worse the geoddb performance
- Why split into feature classes?
  - Each group of features requires distinct custom behavior
  - When feature attributes are different
  - If you need different access privileges for different classes
  - Some features are to use versioning and some not
Storing data in tables

• All objects, features, and relationships are stored in tables

• Predefined fields
  – For identifying objects and storing feature shapes
  – Managed by ArcInfo and should NEVER be modified via another RDBMS

• Custom fields
  – All others
Adding simple behavior with subtypes (p. 89)

- Primarily used to verify the integrity of objects
- Subtypes (lightweight subdivision)
  - Can name them
  - Can define distinct attribute domains
    - Constraints on data values
    - Split and merge policies
  - Can define distinct default values
  - Can define distinct validation rules
    - Attribute rules, connectivity rules, relationship rules
  - Can prescribe relationship types
  - With programming, you can add custom rules
Relationships among objects

- **Topological**
  - Connectivity, adjacency
- **Spatial**
  - Touches, coincides with, overlaps, is inside of, is outside of
- **General**
  - Persistent tie between objects/features