Lecture 2: Object modeling and geodatabases

Georelational data model
- Spatial data is combined with attribute data
  - Spatial data indexed in binary files
  - Attribute data stored in separate tables and linked with a common field
- Topological relationships are stored
- Allowed for customization of attributes
- Spatial data not stored in RDBMS because of limitations in hardware and software

Disadvantage of georelational data model
- All features have the same generic behavior
  - Roads: crossing lines should form a traffic intersection (except over/underpass)
  - Streams: downstream lines should sum the flows of upstream lines

Geodatabase data model
- Allows features to have natural behaviors
- Establish defined relationships between features
- Brings a physical data model closer to a logical data model
  - Owners, buildings, parcels, roads
- Allows custom features to be implemented without writing specialized code

Uses of object behavior
- Adding and editing features
  - Attribute values must fall within a range
  - Adjacency and connectivity tests
  - Conform to a spatial arrangement
  - Geometry matches a set of rules
- Relationships among features
  - Networks should be connected
  - Relationships between features on the map and those not on the map

CAD Data Model
- 1960s and 1970s
- Geographic data stored as points, lines, and areas
- No attributes; each feature type stored on a different layer
- No topology – all “spaghetti”

Geodatabase data model
- Allows features to have natural behaviors
- Establish defined relationships between features
- Brings a physical data model closer to a logical data model
  - Owners, buildings, parcels, roads
- Allows custom features to be implemented without writing specialized code

Uses of object behavior
- Adding and editing features
  - Attribute values must fall within a range
  - Adjacency and connectivity tests
  - Conform to a spatial arrangement
  - Geometry matches a set of rules
- Relationships among features
  - Networks should be connected
  - Relationships between features on the map and those not on the map
Uses of object behavior II

- Cartographic display
  - Auto labeling of contour lines
  - Roads at a large scale are drawn as polygons with clean intersections
  - Coincident lines displayed next to each other
- Interactive analysis
  - Touch a feature and launch an update wizard
  - Select a link and automatically select all downstream links

Benefits of the geodatabase data model

- Uniform repository of geographic data
- Data entry and editing is more accurate
- Users work with more intuitive data objects
- Features have a richer context
- Better maps can be made
- Features on a map display are dynamic
- Shapes of features are better defined
- Sets of features are continuous
- Many users can edit geographic data simultaneously

Within a geodatabase...

- Feature dataset
  - Spatial reference
  - Object (non-spatial) classes
  - Feature (spatial) classes
  - Relationship classes
  - Geometric networks
  - Planar topologies
- Domains
- Validation rules
- Raster datasets
- TIN datasets
- Locators

Key components of OO

- Polymorphism
  - Behaviors of an object class can adapt to variations of objects
- Encapsulation
  - An object is accessed through well-defined software methods, hiding the internal details
- Inheritance
  - An object class can be defined to include the features of another object class, plus additional behaviors

Serving geographic data

- All data is stored in an RDBMS through enhancement
- How does Geodatabase extend an RDBMS
  - Represents 4 types of geo data
  - Stores shapes of features, coordinate systems
  - Can model topologically integrated datasets
  - Can define relationships between objects and features
  - Can enforce attribute integrity
  - Can bind natural feature behavior to feature tables
  - Can present multiple versions of data
- Personal vs. multiuser geodatabases
  - ArcInfo vs. ArcSDE

Building a data model

- Designing a logical data model
- Representing logical data model
- Implementing physical database model
Classes and Objects

- Abstract class
  - Cannot be used to create new objects
  - Is a specification for subclasses

- Creatable class
  - Objects you can directly create

- Instantiable class
  - Cannot directly create new objects
  - Objects created as functions of other objects

Types of Relationships

- Associations – relationships between classes
  - Multiplicity – constraint on associations
    - 1:1 – 1 and only 1 (default)
    - 0..1 – Zero or one
    - M..N – From M to N (positive integers)
    - * or 0..* – From zero to any positive integer
    - 1..* – From 1 to any positive integer

- Type inheritance – classes that share properties with superclass and have additional properties

- Instantiation – one object from one class can create an object from another class

- Aggregation – asymmetric; object of one class is "whole" another is "parts"

- Composition – stronger aggregation; objects from the "whole" class controls the lifetime of the "parts" class