**Conceptual Modeling**

**GIS Applications**
**Spring 2010**

---

**Data models**

Formal representation of something that needs to be understood, remembered, communicated, tested

Represents how users reality is organized in terms of objects, properties, relationships, and processes.

---

**Levels of Data modeling**

- **Real World**
  - Subset of reality
  - Universe of Discourse
  - Universe of modelled phenomena

- **External Level**
  - User defined subset of the real world

- **Conceptual Level**
  - Schematic representation of phenomena and how they are related

- **Logical Level**
  - Transformation of the conceptual model to an implementation model

- **Internal Level**
  - Storage device levels, file structures

---

**Conceptual model**

A formal model in which every entity being modeled in the real world has a corresponding object in the model.

Describes the semantics of some phenomena and represents a series of assertions about its nature.

Describes the things of significance to an organization (entity classes), about which it is inclined to collect information, characteristics (attributes) of these entities, and associations between pairs of entities (relationships).
Conceptual modeling

- A formal analysis and design method that uses a set of guidelines and rules to capture the semantics of a domain
- A conceptual model is built from a limited but well-defined set of constructs
- Constructs, along with a notation, and a small set of rules constitute a formal language or formalism.
- Formal methods include textual or graphical notations to create, present, validate, and manipulate data models
- Clarifies identification of entities, attributes, and relationships
- Provide a basis for discussion and refinement

Creating conceptual models

- Early on formal models were made manually
- Computer Aided Software Engineering (CASE) tools have helped to automate this process
- CASE tools provide formal constructs, graphic representations, constraints, drawing, and editing tools

Basic Components

Entities, Attributes, Associations

Entitiy - principal data objects about which information is collected - usually person, place, thing, event
A particular occurrence of an entity is called an entity instance
Attributes - characteristics of entities or relationships that provide descriptive details about them
Associations - represent real world associations among one or more entities

OO approaches and UML

UML - Unified Modeling language

- models objects, encompassing properties (attributes)
- models relations between objects
- models aggregation of objects into more complex objects
- models generalization or specialization of the types of objects to more general or more specific types

**Representation of a class**

An entity class is represented as a rectangle containing three compartments stacked vertically:

- The top compartment shows the class name.
- The middle compartment lists the class’s attributes.
- The bottom compartment lists the class’s operations or methods.

<table>
<thead>
<tr>
<th>Class name</th>
<th>Lake</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute 1: integer</td>
<td>Name: string</td>
</tr>
<tr>
<td>Attribute 2: string</td>
<td>ID: integer</td>
</tr>
<tr>
<td>Attribute 3: date</td>
<td>pH: float</td>
</tr>
<tr>
<td>Operation 1(): string</td>
<td>Get area(): float</td>
</tr>
</tbody>
</table>

**Associations**

- **Associations** are described in terms of degree, multiplicity and role.
- The degree of a relationship is the number of entities associated in the relationship.
- Unary, binary, ternary are special cases where the degree is 1, 2 or 3.
- An n-ary relationship is the general form for any degree n.
- **Binary** relationships are the most common.
- A ternary relationship relates 3 entities to each other in such a way that it cannot be decomposed into equivalent binary relationships.

**Unary Relation**

- **Stream** flows through **TributaryOf**

**Binary Relation**

- **Stream** flows through **Lake**
  - **hasFlowThrough**

**Ternary Relation**

- **Student** takes offers **Professor**
  - **Course**

**Label** describes the association, typically consists of a noun or verb followed by a preposition.

**Role** indicates a role played by the class in the association. The UML notation expresses the role using a noun on each side of the association.
**Associations**

**Multiplicity** describes the constraint on the number of entity instances that are related through an association.

- It is important to determine the multiplicity on each side of a relationship individually.

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Meaning</th>
<th>Multiplicity A</th>
<th>Multiplicity B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0..1</td>
<td>Zero or one</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0..*</td>
<td>Zero or more</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1..*</td>
<td>One or more</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>n</td>
<td>Only n (where n &gt; 1)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0..n</td>
<td>Zero to n (where n &gt; 1)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1..n</td>
<td>One to n (where n &gt; 1)</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**Associations determined from semantic rules**

- Each department may be the employer of 0 or more employees.
- Each employee may work for at most one department.

**Modeling Inheritance**

Indicated by a solid line drawn from subclass with a closed arrowhead pointing to the super class.
**Associations**

**Composition**: indicates that one class is part of another class, child instance lifespan is dependent on the parent, indicated by a solid line and filled diamond on the parent class.

![Diagram of Composition](image)

**Aggregation**: The child class can exist independently of the parent class. Indicated by a solid line with an unfilled diamond on the parent class.

![Diagram of Aggregation](image)

**Example**

### Entities

<table>
<thead>
<tr>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>parcel</td>
</tr>
<tr>
<td>owner</td>
</tr>
<tr>
<td>building</td>
</tr>
<tr>
<td>permit</td>
</tr>
<tr>
<td>zone</td>
</tr>
</tbody>
</table>

### Attributes

- ID, assessed value, size
- Name, billing address, phone
- ID, assessed value, size, construction date
- Number, type, date issued
- Type, permitted uses

**Example**

**Parcel – owner association**

- A parcel must have at least one owner
- A parcel can have more than one owner
- Owner can own one or more parcels

![Diagram of Parcel Owner](image)

**Example**

**Parcel – building association**

- A parcel may have zero or more buildings
- A building can occupy only one parcel

![Diagram of Parcel Building](image)
Example
Parcel–zone association
- A parcel belongs to only one zone
- A zone contains one to many parcels

Example
Parcel–permit association
- A parcel can have zero to many permits
- An instance of a permit is granted to only one parcel

Example
Building–permit association
- A building can have zero to many permits
- An instance of a permit is associated with only one building
Example
Spatial databases need to consider entities and their spatial representations

<table>
<thead>
<tr>
<th>Parcel</th>
<th>Polygon</th>
</tr>
</thead>
</table>

Spatial Object Definitions - SDTS

Zero-Dimensional Objects

2.3.1.1 Point (NP).
A zero-dimensional object that specifies geometric location. One coordinate pair or triplet specifies the location

2.3.1.2 Node (NO, NN).
A zero-dimensional object that is a topological junction of two or more links or chains, or an end point of a link or chain.

http://mcmcweb.er.usgs.gov/sdts/SDTS_standard_nov97/part1b10.html

Spatial Object Definitions - SDTS

One-Dimensional Objects

2.3.2.1 Line Segment.
A direct line between two points.

2.3.2.2 String (LS).
A connected non-branching sequence of line segments specified as the ordered sequence of points between those line segments. Note: A string may intersect itself or other strings.

2.3.2.3 Arc (AC, AE, AU, AB).
A locus of points that forms a curve that is defined by a mathematical expression.

http://mcmcweb.er.usgs.gov/sdts/SDTS_standard_nov97/part1b10.html
Spatial Object Definitions - SDTS

One-Dimensional Objects continued

2.3.2.4 Link (LQ)
A topological connection between two nodes. A link may be directed by ordering its nodes.

2.3.2.5 Chain.
A directed non-branching sequence of non-intersecting line segments and (or) arcs bounded by nodes, not necessarily distinct, at each end.

2.3.2.5.1 Complete chain (LE).
A chain that explicitly references left and right polygons and start and end nodes. It is a component of a two-dimensional manifold.

2.3.2.6 Ring.
A sequence of nonintersecting chains or strings and (or) arcs, with closure. A ring represents a closed boundary, but not the interior area inside the closed boundary.

http://mcmcweb.cr.usgs.gov/sdts/SDTS_standard_nov97/part1b10.html

Spatial Object Definitions - SDTS

One-Dimensional Objects continued

2.3.2.6.1 G-ring (RS, RA, RM).
A ring created from strings and (or) arcs.

2.3.2.6.2 GT-ring (RU).
A ring created from complete and (or) area chains.

http://mcmcweb.cr.usgs.gov/sdts/SDTS_standard_nov97/part1b10.html

Example –semantic rules

GT_Polygon–Chain association
- A polygon is bound by one or more chains
- An instance of a chain bounds 1 or 2 GT_polygons

Chain-Node association
- A chain is bound by one to two nodes
- A node bounds 1 or more chains

Line_segment-Chain association
- A chain contains 1 or more line segments
- An instance of a line_segment is part of one and only one chain.
Example – semantic rules

**Line segment-point association**
- A line segment is bound by 2 points.
- An instance of a point bounds one or more line segments.

**Node-point association**
- A node can be associated with one point.
- An instance of a point can be associated with at most one node.

---

Example – Geometry and Topology

![Geometry and Topology Diagram]

---

**Logical model**

The logical model organizes data in terms of a particular data management technology.

Given current predominance of relational databases, logical model generally conforms to relational theory - contains **fully normalized entities**.

For a logical data model to be normalized, it must include the **full population of attributes** and those attributes must be defined in terms of their **domains** or **logical data types** (e.g., character, number, date, pict)

---

**Logical Modeling**

Verify model logic, coherence, and completeness of the model

Convert conceptual model to a specific implementation model - relational model

<table>
<thead>
<tr>
<th>Entities</th>
<th>Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>polygon</td>
<td>polygon (Polygon-ID, area)</td>
</tr>
<tr>
<td>chain</td>
<td>Chain (Chain-ID, length)</td>
</tr>
<tr>
<td>node</td>
<td>Node (node-ID)</td>
</tr>
</tbody>
</table>
Logical Modeling

Entities converted to relations

- **Line segment**
  - Line-Segment (line_segment-ID, length)
- **point**
  - Point (point-ID, x, y)

Logical Modeling

Associations converted to relations

- **polygon**
  - 1..2 bound by bounds 1..*
  - Polygon-chain(polygon-ID, chain-ID)

<table>
<thead>
<tr>
<th>Polygon-ID</th>
<th>Chain-ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>a</td>
</tr>
<tr>
<td>101</td>
<td>b</td>
</tr>
<tr>
<td>102</td>
<td>c</td>
</tr>
<tr>
<td>102</td>
<td>d</td>
</tr>
<tr>
<td>102</td>
<td>e</td>
</tr>
<tr>
<td>103</td>
<td>e</td>
</tr>
<tr>
<td>103</td>
<td>f</td>
</tr>
<tr>
<td>104</td>
<td>g</td>
</tr>
</tbody>
</table>

Logical Modeling

- **chain**
  - 1..* bound by bounds 2..2
  - Chain-node(chain-ID, from-node-ID, to-node-ID)

<table>
<thead>
<tr>
<th>Chain-ID</th>
<th>From-Node-ID</th>
<th>To-Node-ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>b</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>c</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>d</td>
<td>10</td>
<td>40</td>
</tr>
<tr>
<td>e</td>
<td>30</td>
<td>40</td>
</tr>
<tr>
<td>f</td>
<td>30</td>
<td>40</td>
</tr>
<tr>
<td>g</td>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>

Logical Modeling

- **chain**
  - 1..1 contains is part of 1..*
  - Chain-Line_Segment(chain-ID, line_segment-ID)

<table>
<thead>
<tr>
<th>Chain-ID</th>
<th>Line_segment-ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>3a</td>
</tr>
<tr>
<td>a</td>
<td>3a</td>
</tr>
<tr>
<td>b</td>
<td>3a</td>
</tr>
<tr>
<td>c</td>
<td>3a</td>
</tr>
<tr>
<td>d</td>
<td>3a</td>
</tr>
<tr>
<td>e</td>
<td>3a</td>
</tr>
<tr>
<td>f</td>
<td>3a</td>
</tr>
<tr>
<td>g</td>
<td>3a</td>
</tr>
</tbody>
</table>
Logical Modeling

<table>
<thead>
<tr>
<th>Line_Segment_ID</th>
<th>Start_Point_ID</th>
<th>End_Point_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>ls1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>ls2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>ls3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>ls4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>ls5</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>ls6</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>ls7</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>ls8</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>ls9</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>ls10</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>ls11</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>ls12</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>ls13</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>ls14</td>
<td>14</td>
<td>15</td>
</tr>
</tbody>
</table>

Logical Modeling

<table>
<thead>
<tr>
<th>Node-ID</th>
<th>Point-ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>30</td>
<td>6</td>
</tr>
<tr>
<td>40</td>
<td>7</td>
</tr>
<tr>
<td>50</td>
<td>10</td>
</tr>
</tbody>
</table>