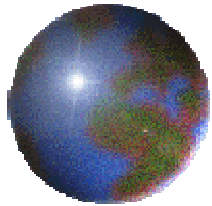




Chapter 2: Spatial Concepts and Data Models



- 2.1 Introduction
- 2.2 Models of Spatial Information
- 2.3 Three-Step Database Design
- 2.4 Extending ER with Spatial Concepts
- 2.5 Summary



What is a Data Model?

- What is a model? (dictionary meaning)
 - A set of plans (blueprint drawing) for a building
 - A miniature representation of a system to analyze properties of interest
- What is Data Model?
 - Specify structure or schema of a data set
 - Document description of data
 - Facilitates early analysis of some properties, e.g. querying ability, redundancy, consistency, storage space requirements, etc.
- Examples:
 - GIS organize spatial set as a set of layers
 - Databases organize dataset as a collection of tables



Why Data Models?

- Data models facilitate
 - Early analysis of properties, e.g. storage cost, querying ability, ...
 - Reuse of shared data among multiple applications
 - Exchange of data across organization
 - Conversion of data to new software / environment
- Example- Y2K crisis for year 2000

Many computer software systems were developed without well-defined data models in 1960s and 1970s. These systems used a variety of data models for representing time and date. Some of the representations used two digits to represent years. In late 1990s, people worried that the 2 digit representation of year may lead to erroneous behavior. For example age of a person born in 1960 (represented as 60) in year 2000 (represented as 00) may appear negative and may be flagged as illegal data item. A large amount of effort and resources (hundreds of Billions of dollars) was spent in revising the software. Proper use of data model may have significantly reduced the costs. If time and date were modeled as abstract data types in a software, only a small portion of the software implementing the date ADT had to be reviewed and revised.



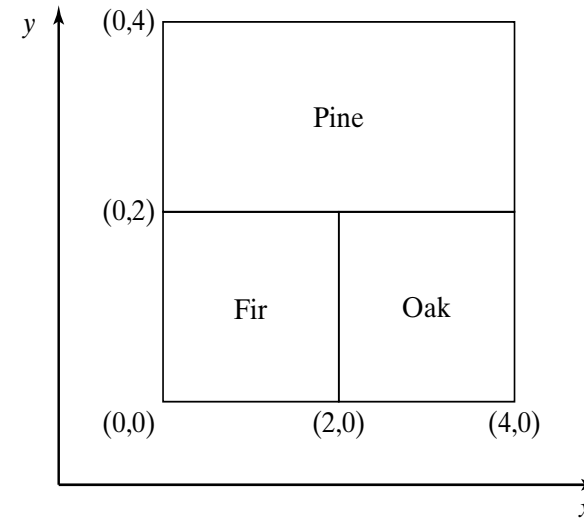
Types of Data Models

- Two Types of data models
 - Generic data models
 - Developed for business data processing
 - Support simple abstract data types (ADTs), e.g. numbers, strings, date
 - Not convenient for spatial ADTs, e.g. polygons
 - Recall a polygon becomes dozens of rows in 3 tables (Fig.1.4, pp.8)
 - Need to extend with spatial concepts, e.g. ADTs
 - Application domain specific, e.g. spatial models
 - Set of concepts developed in Geographic Information Science
 - Common spatial ADTs across different GIS applications
- Plan of Study
 - First study concepts in spatial models
 - Then study generic model
 - Finally put the two together



Models of Spatial Information

- ☉ Two common models
 - ▣ Field based
 - ▣ Object based
- ☉ Example: Forest stands
 - a. forest stand map
 - b. object view has 3 polygons
 - c. field view has a function



(a)

Object Viewpoint of Forest Stands

Area-ID	Dominant Tree Species	Area/Boundary
FS1	Pine	[(0,2),(4,2),(4,4),(0,4)]
FS2	Fir	[(0,0),(2,0),(2,2),(0,2)]
FS3	Oak	[(2,0),(4,0),(4,2),(2,2)]

(b)

Field Viewpoint of Forest Stands

$$f(x,y) = \begin{cases} \text{"Pine,"} & 2 \leq x \leq 4; 2 < y \leq 4 \\ \text{"Fir,"} & 0 \leq x \leq 2; 0 \leq y \leq 2 \\ \text{"Oak,"} & 2 < x \leq 4; 0 \leq y \leq 2 \end{cases}$$

(c) 5

Field Based Model

- ☉ Three main concepts:
 - ☒ Spatial Framework is a partitioning of space
 - e.g., Grid imposed by Latitude and Longitude
 - ☒ Field Functions:
f: Spatial Framework → Attribute Domain
 - ☒ Field Operations
 - e.g., addition (+) and composition (o)

$$f + g : x \rightarrow f(x) + g(x)$$

$$f \circ g : x \rightarrow f(g(x))$$



Types of Field Operations

- ⊕ **Local:** value of the new field at a given location in the spatial frame-work depends only on the value of the input field at that location(e.g., thresholding)
- ⊕ **Focal:** value of the resulting field at a given location depends on the values that the input field assumes in a small neighborhood of the location (e.g., gradient)
- ⊕ **Zonal:** Zonal operations are naturally associated with aggregate operators or the integration function. An operation that calculates the average height of the trees for each species is a zonal operation
- ⊕ **Exercise:** Classify following operations on elevation field
 - I. Identify peaks (points higher than its neighbors)
 - II. Identify mountain ranges (elevation over 2000 feet)
 - III. Determine average elevation of a set of river basins



Object Model

🌀 Object model concepts

- ❏ Objects: distinct identifiable things relevant to an application
- ❏ Objects have attributes and operations
- ❏ Attribute: a simple (e.g. numeric, string) property of an object
- ❏ Operations: function maps object attributes to other objects

🌀 Example from a roadmap

- ❏ Objects: roads, landmarks, ...
- ❏ Attributes of road objects:
 - spatial: location, e.g. polygon boundary of land-parcel
 - non-spatial: name (e.g. Route 66), type (e.g. interstate, residential street), number of lanes, speed limit, ...
- ❏ Operations on road objects: determine center line, determine length, determine intersection with other roads, ...



Classifying Spatial Objects

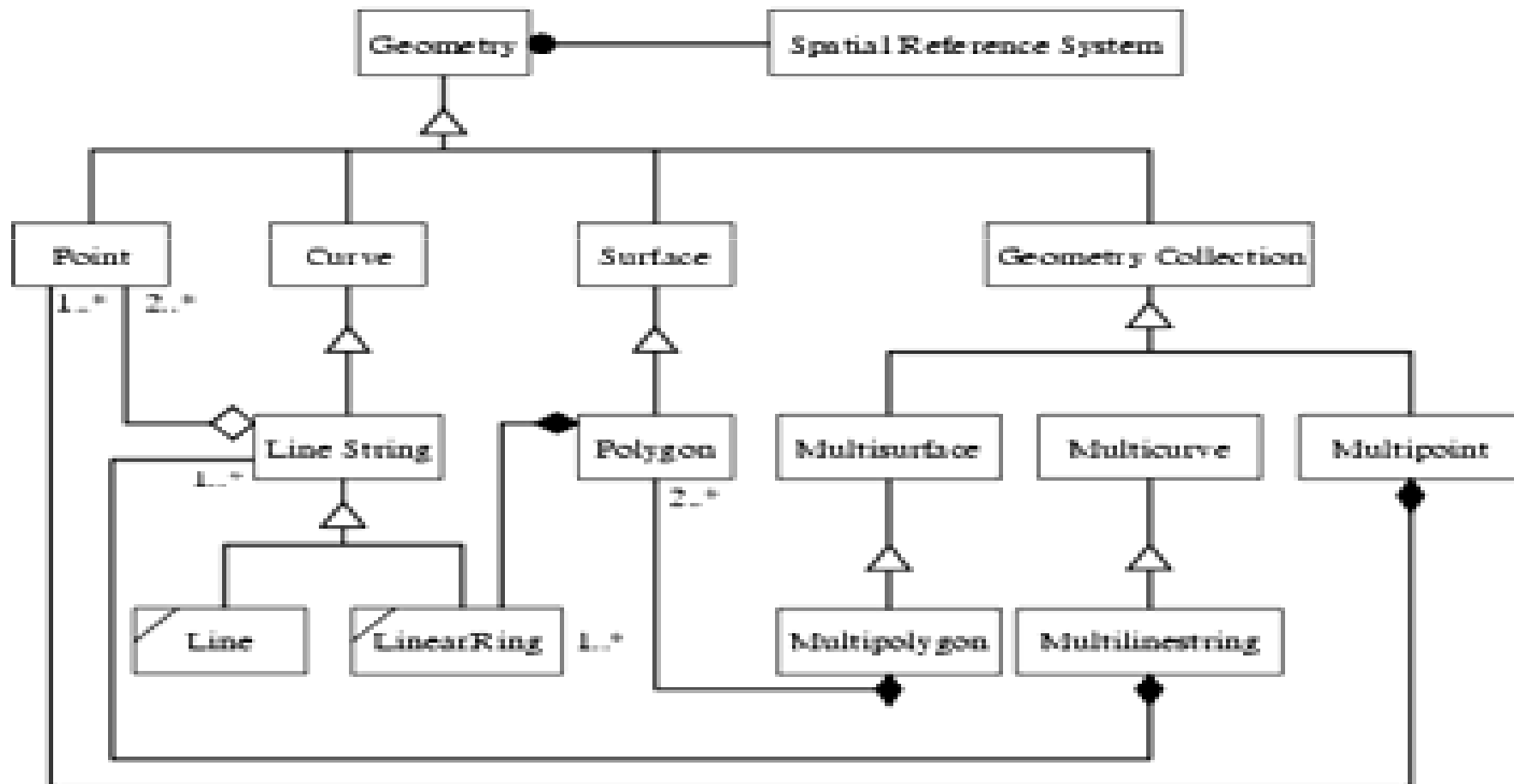
- Spatial objects are spatial attributes of general objects
- Spatial objects are of many types
 - Simple
 - 0-dimensional (points), 1-dimensional (curves), 2-dimensional (surfaces)
 - Example given at the bottom of this slide
 - Collections
 - Polygon collection (e.g. boundary of Japan or Hawaii), ...
 - See more complete list in Figure 2.2

Spatial Object Types	Example Object	Dimension
Point	City	0
Curve	River	1
Surface	Country	2



Spatial Object Types in OGIS Data Model

Each rectangle shows a distinct spatial object type





Classifying Operations on Spatial Objects in Object Model

- Classifying operations (Tables 2.1, 2.2, pp.29-31)
 - Set based: 2-dimensional spatial objects (e.g. polygons) are sets of points
 - a set operation (e.g. intersection) of 2 polygons produce another polygon
 - Topological operations: boundary of USA touches boundary of Canada
 - Directional: New York city is to east of Chicago
 - Metric: Chicago is about 700 miles from New York city
 - Q? Identify classes of spatial operations not listed in this slide

Set theory based	Union, Intersection, Containment
Topological	Touches, Disjoint, Overlap, etc.
Directional	East, North-West, etc.
Metric	Distance



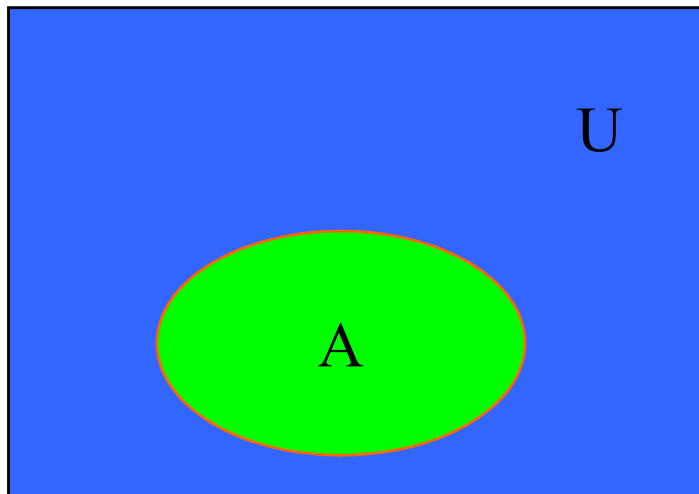
Topological Relationships

- Topological Relationships
 - Invariant under elastic deformation (without tear, merge)
 - Two countries which touch each other in a planar paper map will continue to do so in spherical globe maps
- Topology is the study of topological relationships
- Example queries with topological operations
 - What is the topological relationship between two objects A and B ?
 - Find all objects which have a given topological relationship to object A ?



Topological Concepts

- Interior, boundary, exterior
 - Let A be an object in a “Universe” U



Green is A interior (A^o)

Red is boundary of A (∂A)

Blue—(Green+Red) is A exterior (A^-)

- Question: Define Interior, boundary, exterior on curves and points



Nine-Intersection Model of Topological Relationships

- Many topological Relationship between A and B can be
 - specified using 9 intersection model
 - examples on next slide
- Nine intersections
 - intersections between interior, boundary, exterior of A, B
 - A and B are spatial objects in a two dimensional plane.
 - Can be arranged as a 3 by 3 matrix
 - Matrix element take a value of 0 (false) or 1 (true)
- Q? Determine the number of many distinct 3 by 3 Boolean matrices

$$\Gamma_9(A, B) = \begin{pmatrix} A^\circ \cap B^\circ & A^\circ \cap \partial B & A^\circ \cap B^- \\ \partial A \cap B^\circ & \partial A \cap \partial B & \partial A \cap B^- \\ A^- \cap B^\circ & A^- \cap \partial B & A^- \cap B^- \end{pmatrix}$$



Specifying Topological Operation in 9-Intersection Model

9 intersection matrices for a few topological operations

$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ <p>disjoint</p>	$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$ <p>contains</p>	$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$ <p>inside</p>	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ <p>equal</p>
$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ <p>meet</p>	$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$ <p>covers</p>	$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$ <p>coveredBy</p>	$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ <p>overlap</p>

Question: Can this model specify topological operation between a polygon and a curve?



Using Object Model of Spatial Data

- Object model of spatial data
 - OGIS standard set of spatial data types and operations
 - Similar to the object model in computer software
 - Easily used with many computer software systems
 - Programming languages like Java, C++, Visual basic
 - Example use in a Java program is in section 2.1.6
 - Post-relational databases, e.g. OODBMS, ORDBMS
 - Example usage in chapter 3 through 6



Three-Step Database Design

- ⊗ Database applications are modeled using a three-step design process
 - ⊠ Conceptual-datatypes, relationships and constraints (ER model)
 - ⊠ Logical-mapping to a Relational model and associated query language (Relational Algebra)
 - ⊠ Physical-file structures, indexing, ...
- ⊗ Scope
 - ⊠ We discuss conceptual and logical data models in section 2.3
 - ⊠ Physical model is discussed in chapter 4

Example Application Domain

- ✿ Database design is for a specific application domain
 - ▣ Often a requirements document is available
 - ▣ Designers discuss requirements with end-users as needed
 - ▣ We will use a simple spatial application domain
 - to illustrate concepts in conceptual and logical data models
 - to illustrate translation of conceptual DM to logical DM
- ✿ Spatial application domain
 - ▣ A *state-park* consists of *forests*.
 - ▣ A forest is a collection of *forest-stands* of different species
 - ▣ State-Park is accessed by *roads* and has a *manager*
 - ▣ State-Park has *facilities*
 - ▣ *River* runs through state-park and supplies water to the facilities

Conceptual DM: The ER Model

☉ 3 basic concepts

- ☒ Entities have an independent conceptual or physical existence
 - Examples: Forest, Road, Manager, ...
- ☒ Entities are characterized by Attributes
 - Example: Forest has attributes of name, elevation, etc.
- ☒ An Entity interacts with another Entity through relationships.
 - Road allow access to Forest interiors
 - This relationship may be name “Accesses”

☉ Comparison with Object model of spatial information

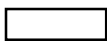



- ☒ Entities are collections of attributes are like objects
- ☒ However ER model does not permit general user defined operations
- ☒ Relationships are not directly supported in Object model
 - but may be simulated via operations

Relationship Types

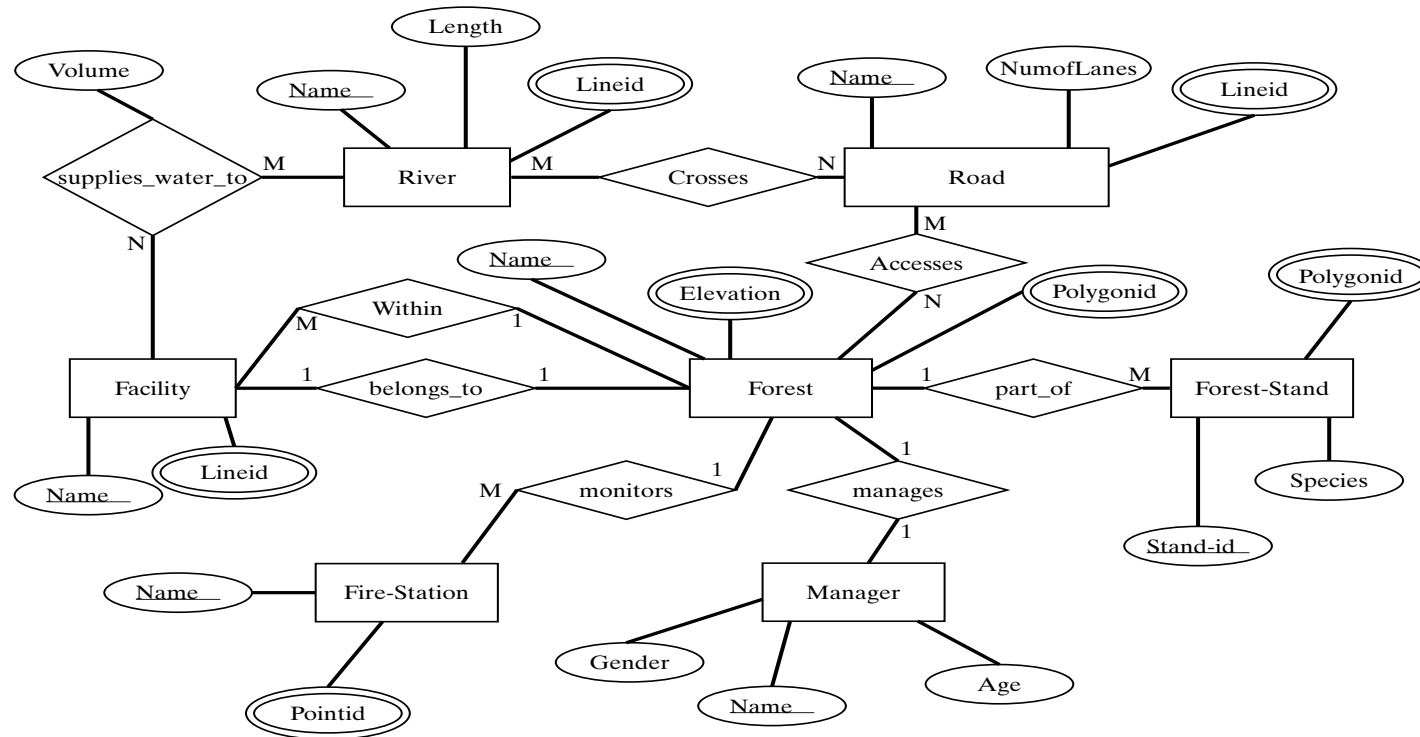
- ☉ Relationships can be categorized by
 - ☒ cardinality constraints
 - ☒ other properties, e.g. number of participating entities
 - Binary relationship: two entities participate
- ☉ Types of Cardinality constraints for binary relationships
 - ☒ One-One: An instance of an entity relates to a unique instance of other entity
 - ☒ Many-One: Many instances of an entity relate to an instance of an other
 - ☒ Many-Many: Many instances of one entity relate to multiple instances of another
- ☉ Exercise: Identify type of cardinality constraint for following:
 - ☒ Many facilities belong to a forest. Each facility belong to one forest
 - ☒ A manager manages 1 forest. Each forest has 1 manager
 - ☒ A river supplies water to many facilities. A facility gets water from many rivers

ER Diagrams Graphical Notation

- ER Diagrams are graphic representation of ER models
 - Several different graphic notation are used
 - We use a simple notation summarized below
 - Example ER Diagram for Forest example in next slide
- Q? Compare and contrast “Attributes” and “Multi-valued attributes”

Concept	Symbol
Entities	
Attributes	
Multi-valued Attributes	
Relationships	
Cardinality of Relationship	1:1, M:1, M:N

ER Diagram for “State-Park”



- Exercise:
 - List the entities, attributes, relationships in this ER diagram
 - Identify cardinality constraint for each relationship
 - How many roads “Accesses” a “Forest_stand”? (one or many)

Logical Data Model: The Relational Model

- ⊗ Relational model is based on set theory
- ⊗ Main concepts
 - ⊠ Domain: a set of values for a simple attribute
 - ⊠ Relation: cross-product of a set of domains
 - Represents a table, i.e. homogeneous collection of rows (tuples)
 - The set of columns (i.e. attributes) are same for each row
- ⊗ Comparison to concepts in conceptual data model
 - ⊠ Relations are similar to but not identical to entities
 - ⊠ Domains are similar to attributes
 - ⊠ Translation rules establishing exact correspondence are discussed in 2.2.3

Relational Schema

🌀 Schema of a Relation

- ❏ Enumerates columns, identifies primary key and foreign keys
- ❏ Primary Key :
 - one or more attributes uniquely identify each row within a table
- ❏ Foreign keys
 - R's attributes which form primary key of another relation S
 - Value of a foreign key in any tuple of R match values in some row of S

🌀 Relational schema of a database

- ❏ Collection of schemas of all relations in the database
- ❏ Example: Figure 2.5 (next slide)
- ❏ A blue print summary drawing of the database table structures
- ❏ Allows analysis of storage costs, data redundancy, querying capabilities
- ❏ Some databases were designed as relational schema in 1980s
- ❏ Nowadays, databases are designed as ER models and relational schema is generated via CASE tools

Relational Schema Example

- Exercise:
- Identify relations with
 - primary keys
 - foreign keys
 - other attributes
- Compare with ER diagram
 - Figure 2.4, pp.37

Forest-Stand

<u>Stand-id</u>	Species	Forest-name
(Integer)	(varchar)	(varchar)

River

<u>Name</u>	Length
(varchar)	(Real)

Road

<u>Name</u>	NumofLanes
(varchar)	(Integer)

Facility

<u>Name</u>	Forest-name	Forest-name-2
(varchar)	(varchar)	(varchar)

Forest

<u>Name</u>
(varchar)

Fire-Station

<u>Name</u>	ForName
(varchar)	(varchar)

Supplies_Water_To

<u>FacName</u>	<u>RivName</u>	Volume
(varchar)	(varchar)	(Real)

Manager

<u>Name</u>	<u>Age</u>	<u>Gender</u>	ForName
(varchar)	(Integer)	(varchar)	(varchar)

Fstand-Geom

<u>Stand-id</u>	<u>Polygonid</u>
(Integer)	(Integer)

River-Geom

<u>Name</u>	<u>Lineid</u>
(Integer)	(Integer)

Road-Geom

<u>Rname</u>	<u>Lineid</u>
(varchar)	(Integer)

Facility-Geom

<u>Name</u>	<u>Pointid</u>
(varchar)	(Integer)

Forest-Geom

<u>Name</u>	<u>Polygonid</u>
(varchar)	(Integer)

Fstation-Geom

<u>Name</u>	<u>Pointid</u>
(varchar)	(Integer)

Road-Access-Forest

<u>RoadName</u>	<u>ForName</u>
(varchar)	(varchar)

--



Relational Schema for Point, Line, Polygon and Elevation

- Relational model restricts attribute domains
 - simple atomic values, e.g. a number
 - Disallows complex values (e.g. polygons) for columns
 - Complex values need to be decomposed into simpler domains
 - A polygon may be decomposed into edges and vertices (Fig.2.5)

Polygon

<u>Polygonid</u>	<u>Seq-no</u>	Pointid
(Integer)	(Integer)	(Integer)

Line

<u>Lineid</u>	<u>Seq-no</u>	Pointid
(Integer)	(Integer)	(Integer)

Point

<u>Pointid</u>	Latitude	Longitude
(Integer)	(Real)	(Real)

Elevation

<u>Forest-name</u>	Pointid (F.K.)	Elevation
(varchar)	(Integer)	(Real)

Figure 2.5

More on Relational Model

☉ Integrity Constraints

- ☒ Key: Every relation has a primary key
- ☒ Entity Integrity: Value of primary key in a row is never undefined
- ☒ Referential Integrity: Value of an attribute of a Foreign Key must appear as a value in the primary key of another relationship or must be null

☉ Normal Forms (NF) for Relational schema

- ☒ Reduce data redundancy and facilitate querying
- ☒ 1st NF: Each column in a relation contains an atomic value
- ☒ 2nd and 3rd NF: Values of non-key attributes are fully determined by the values of the primary key, only the primary key, and nothing but the primary key
- ☒ Other normal forms exist but are seldom used
- ☒ Translating a well-designed ER model yields a relational schema in 3rd NF
 - satisfying definition of 1st, 2nd and 3rd normal forms

Mapping ER to Relational

- Highlights of translation rules (section 2.2.3)
 - Entity becomes Relation
 - Attributes become columns in the relation
 - Multi-valued attributes become a new relation
 - includes foreign key to link to relation for the entity
 - Relationships (1:1, 1:N) become foreign keys
 - M:N Relationships become a relation
 - containing foreign keys or relations from participating entities
- Example and Exercise
 - Compare Fig.2.4 and Fig.2.5
 - Identify the relational schema components for
 - entity Facility, its attributes and its relationships
 - Note an empty relation box in Fig.2.5. Fill in its schema



Extending ER with Spatial Concepts

- Motivation
 - ER Model is based on discrete sets with no implicit relationships
 - Spatial data comes from a continuous set with implicit relationships
 - Any pair of spatial entities has relationships like distance, direction, ...
- Explicitly drawing all spatial relationship
 - clutters ER diagram
 - generates additional tables in relational schema
 - Misses implicit constraints in spatial relationships (e.g. partition)
- Pictograms
 - Label spatial entities along with their spatial data types
 - Allows inference of spatial relationships and constraints
 - Reduces clutter in ER diagram and relational schema
 - Example: Fig.2.7 (next slide) is simpler than Fig.2.4

ER Diagram with Pictograms: An Example

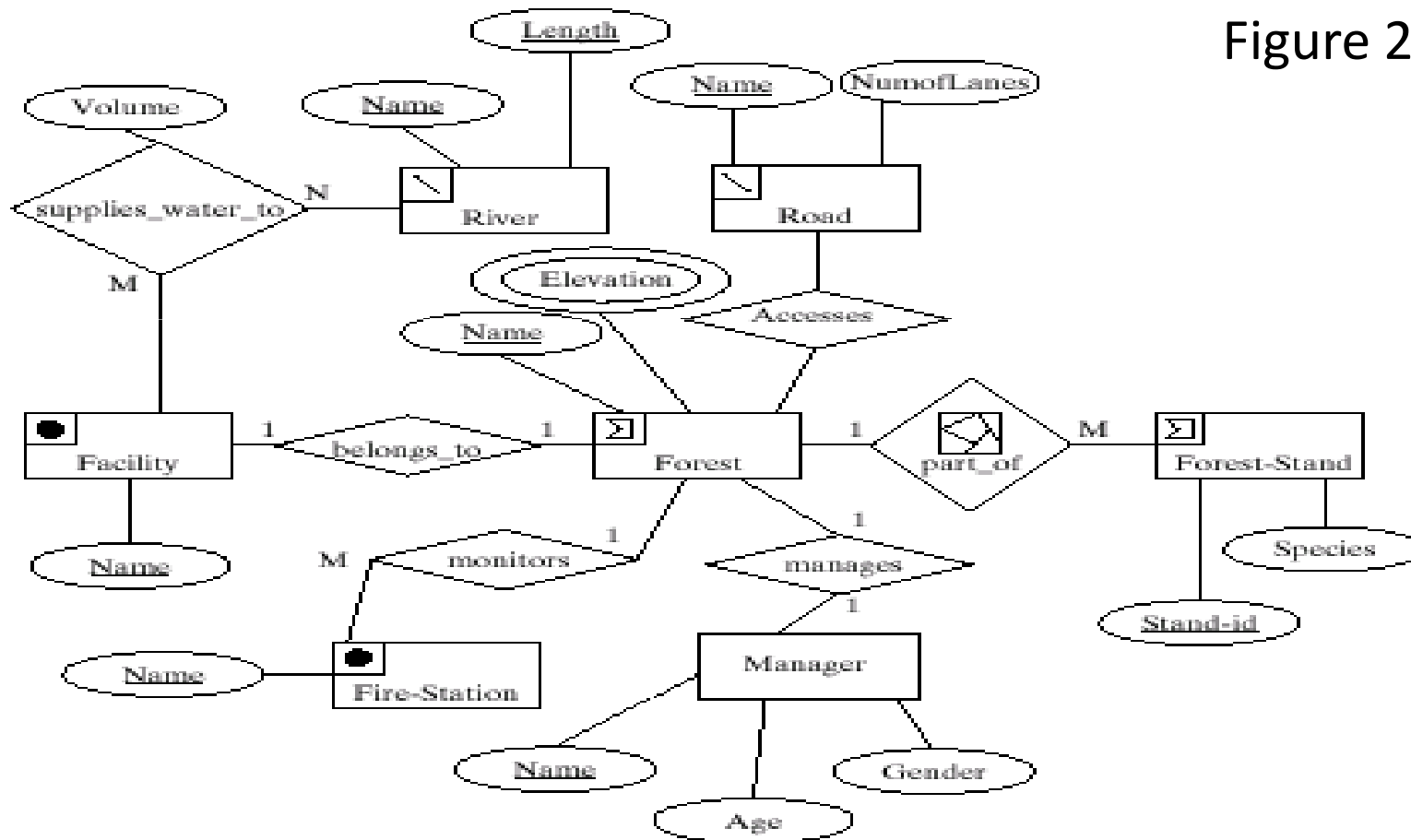
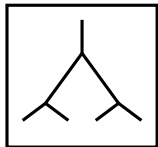


Figure 2.7

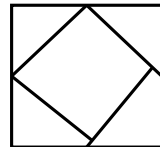


Specifying Pictograms

- Grammar based approach
 - Rewrite rule
 - like English syntax diagrams
- Classes of pictograms
 - Entity pictograms
 - basic: point, line, polygon
 - collection of basic
 - ...
 - Relationship pictograms
 - partition, network

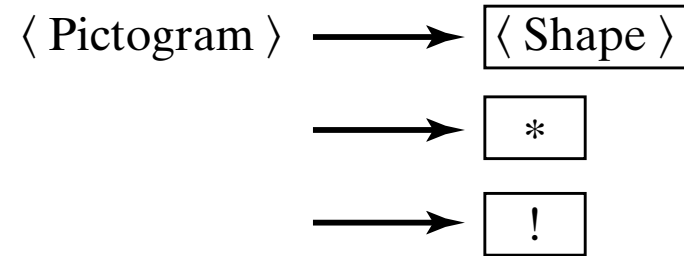


Part_of(Network)

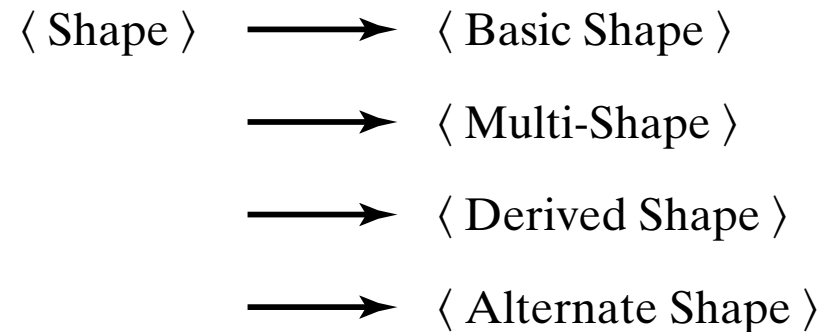


Part_of(Partition)

Pictograms for Relationships



Grammar (for Pictogram)



Grammar (for Shape)

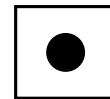


Entity Pictograms: Basic shapes, Collections

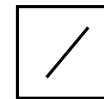
⟨ Basic Shape ⟩ → ●

→ /

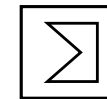
→ ∑



Point



Line



Polygon

Grammar (for Basic Shape)

Pictograms for Basic Shapes

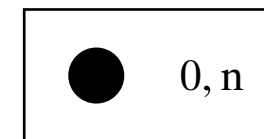
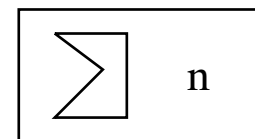
⟨ Cardinality ⟩ → 0, 1

→ 1

→ 1, n

→ 0, n

→ n



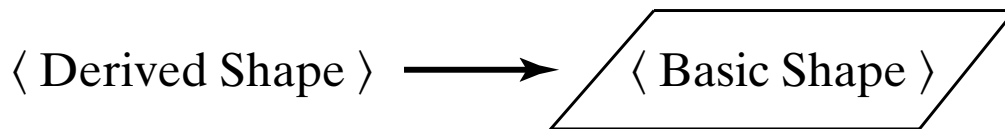
Grammar (for Cardinality)

*Pictograms Multishapes
(using cardinality)*

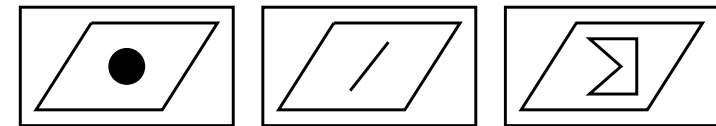


Entity Pictograms: Derived and Alternate Shapes

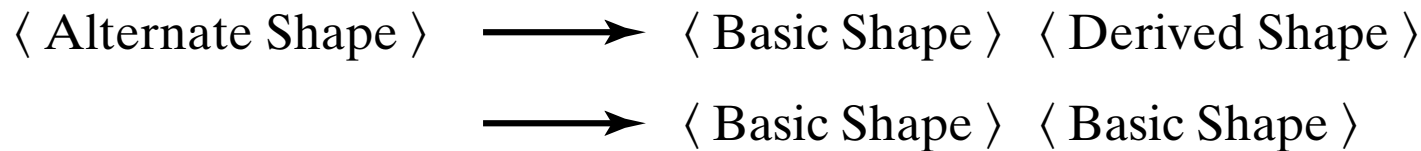
- Derived shape example is city center point from boundary polygon
- Alternate shape example: A road is represented as a polygon for construction or as a line for navigation



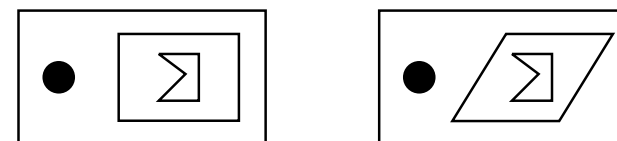
Grammar (for Derived Shape)



Pictograms for Derived Shapes



Grammar (for Alternate Shape)



*Pictograms for Alternate Shapes*³³

Conceptual Data Modeling with UML

- Motivation
 - ER Model does not allow user defined operations
 - Object oriented software development uses UML
 - UML stands for Unified Modeling Language
 - It is a standard consisting of several diagrams
 - class diagrams are most relevant for data modeling
- UML class diagrams concepts
 - Attributes are simple or composite properties
 - Methods represent operations, functions and procedures
 - Class is a collection of attributes and methods
 - Relationship relate classes
- Example UML class diagram: Figure 2.8

UML Class Diagram with Pictograms: Example

- Exercise: Identify classes, attributes, methods, relationships in Fig.2.8
 - Compare Fig.2.8 with corresponding ER diagram in Fig.2.7

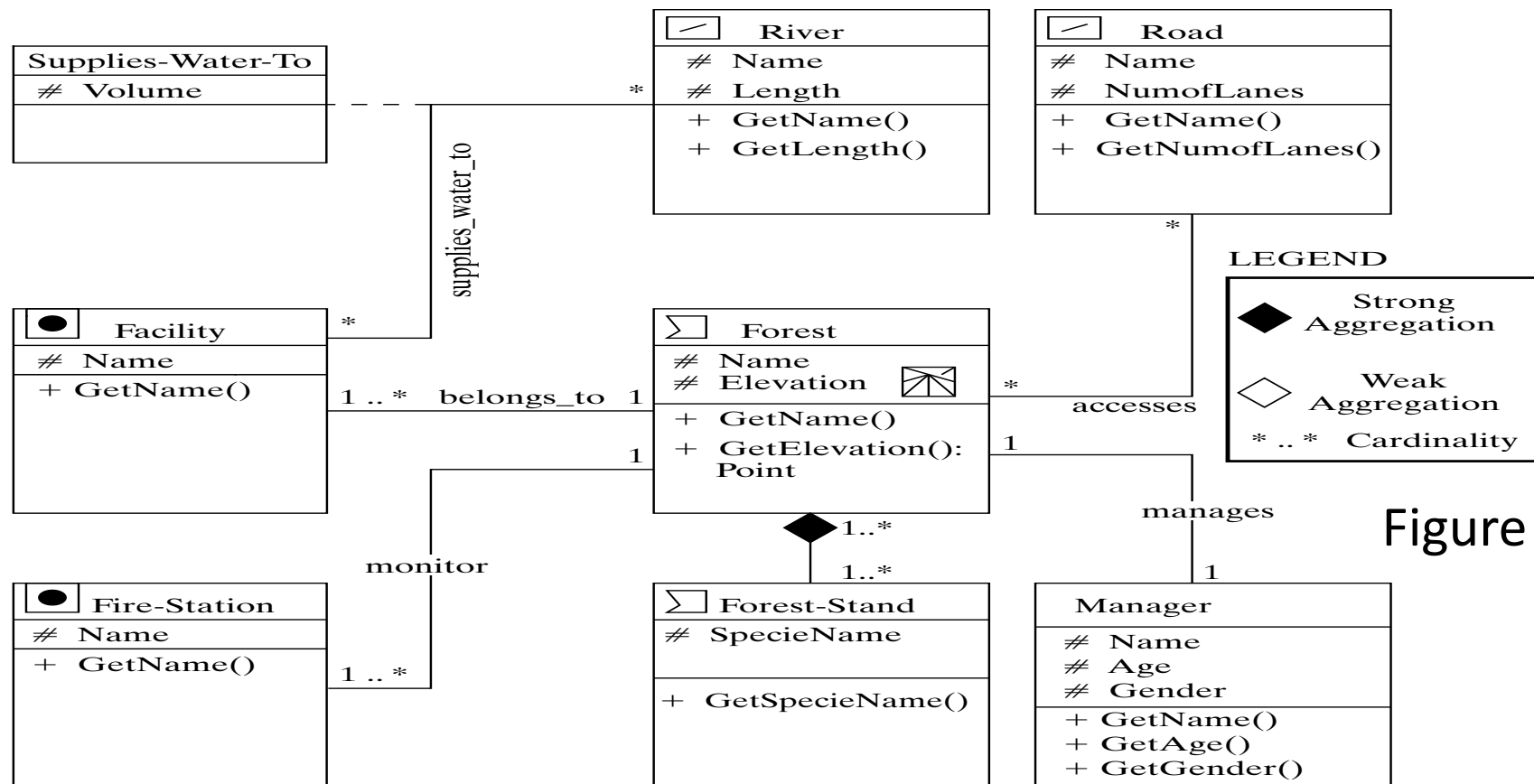


Figure 2.8

Comparing UML Class Diagrams to ER Diagrams

- Concepts in UML class diagram vs. those in ER diagrams
 - Class without methods is an Entity
 - Attributes are common in both models
 - UML does not have key attributes and integrity constraints
 - ER diagram does not have methods
 - Relationships properties are richer in ER diagrams
 - Entities in ER diagrams relate to datasets, but UML class diagram can contain classes, which have little to do with data

Summary

- ⊗ Spatial Information modeling can be classed into Field based and Object based
- ⊗ Field based for modeling smoothly varying entities, like rainfall
- ⊗ Object based for modeling discrete entities, like country
- ⊗ A data model is a high level description of the data
 - ⊗ it can help in early analysis of storage cost, data quality
- ⊗ There are two popular models of spatial information
 - ⊗ Field based and Object based
- ⊗ Database are designed in 3-steps
 - ⊗ Conceptual, Logical and Physical
- ⊗ Pictograms can simplify Conceptual data models