Chapter 2: Spatial Concepts and Data Models

Learning Objectives

- Learning Objectives (LO)
  - LO1: Understand concept of data models
  - LO2 : Understand the models of spatial information
  - LO3: Understand the 3-step design of databases
    - Conceptual - ER model
    - Logical - Relational model
    - Physical
    - Translation from Conceptual to Logical
  - LO4: Learn about the trends in spatial data models

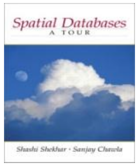
- Mapping chapter sections to learning objectives
  - LO2 - 2.1
  - LO3 - 2.2
  - LO4 - 2.3, 2.4

2.2 Three-Step Database Design

- Database applications are modeled using a three-step design process
  - Conceptual data types, relationships and constraints (ER model)
  - Logical-mapping to a Relational model and associated query language(Relational Algebra)
  - Physical-file structures, indexing

Example Application Domain

- Spatial application domain
  - A *state-park* consists of *forests*.
  - A forest is a collection of *forest-stands* of different species
  - Forests are accessed by *roads*
  - Each forest has a *manager*
  - Forests have *facilities*
  - *Rivers* runs through forests and supplies water to the facilities
  - Forests are monitored by *fire-stations*



2.2.1 Conceptual DM: The ER Model

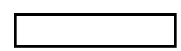
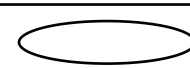
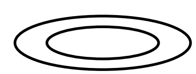
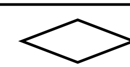
- 3 basic concepts
  - Entities have an independent conceptual or physical existence.
    - Examples: Forest, Road, Manager, ...
  - Entities are characterized by Attributes
    - Example: Forest has attributes of name, elevation, etc.
  - An Entity interacts with another Entity through relationships.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

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.

ER Diagrams Graphical Notation

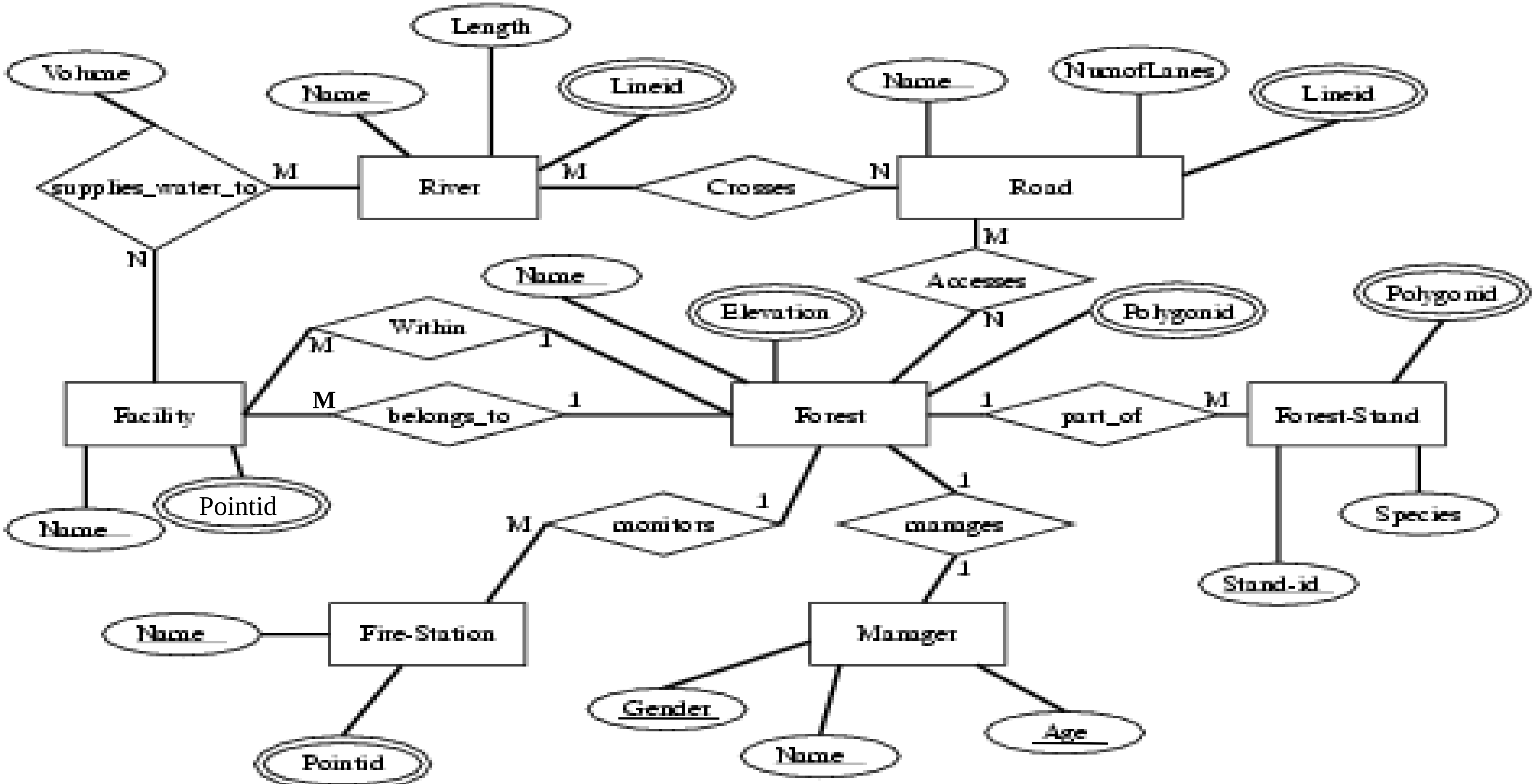
- ER Diagrams are graphic representation of ER models
  - Several different graphic notation are used
  - We use a simple notation summarized below
  - Example ER Diagram for Forest exampl in next slide
- Q? Compare and contrast “Atributes” and “Multi-valued attributes”.

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

Example

- Create the ER diagram for the State-Park example
  - Entities and characteristic attributes
  - Relationships and constraints
  - Both spatial and non spatial

ER Diagram for "State-Park"



2.2.2 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

2.2.3 Mapping ER to Relational

- Highlights of translation rules (section 2.2.3)
  - Entity becomes Relation
  - Attributes become columns in the relation
  - Relationships (1:1, 1:N) become foreign keys
  - M:N Relationships become a relation
    - containing foreign keys or relations from participating entities
  - Multi-valued attributes become a new relation
    - includes foreign key to link to relation for the entity

Example

- Create the relational model of the example ER
  - Relations
  - Keys
  - Spatial types

Relational Schema Example

Forest-Stand

Stand-id	Species	Forest-name
(Integer)	(varchar)	(varchar)

Fstand-Geom

Stand-id	Polygonid
(Integer)	(Integer)

River

Name	Length
(varchar)	(Real)

River-Geom

Name	Lineid
(Integer)	(Integer)

Road

Name	NumofLanes
(varchar)	(Integer)

Road-Geom

Rname	Lineid
(varchar)	(Integer)

Facility

Name	Forest-name	Forest-name-2
(varchar)	(varchar)	(varchar)

Facility-Geom

Name	Pointid
(varchar)	(Integer)

Forest

Name
(varchar)

Forest-Geom

Name	Polygonid
(varchar)	(Integer)

Fire-Station

Name	ForName
(varchar)	(varchar)

Fstation-Geom

Name	Pointid
(varchar)	(Integer)

Supplies_Water_To

FacName	RivName	Volume
(varchar)	(varchar)	(Real)

Road-Access-Forest

RoadName	ForName
(varchar)	(varchar)

Manager

Name	Age	Gender	ForName
(varchar)	(Integer)	(varchar)	(varchar)

Crosses

RivName	RoadName
(varchar)	(varchar)

Similar for Accesses

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

Polygon

Polygonid	Seq-no	Pointid
(Integer)	(Integer)	(Integer)

Line

Lineid	Seq-no	Pointid
(Integer)	(Integer)	(Integer)

Elevation applies to points

Point

Pointid	Latitude	Longitude
(Integer)	(Real)	(Real)

Elevation

Forest-name	Pointid (F.K.)	Elevation
(varchar)	(Integer)	(Real)

Learning Objectives

- Learning Objectives (LO)
  - LO1: Understand concept of data models
  - LO2 : Understand the models of spatial information
  - LO3: Understand the 3-step design of databases
  - LO4: Learn about the trends in spatial data models
    - Pictograms in conceptual models
    - UML class diagrams

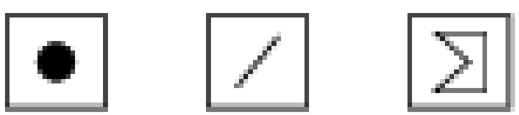
- Mapping Sections to learning objectives
  - LO2 - 2.1
  - LO3 - 2.2
  - LO4 - 2.3, 2.4

2.3 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)

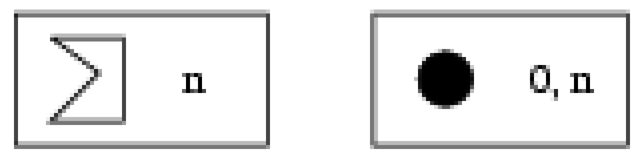
Pictograms

- Label spatial entities along with their spatial data types
- Allows inference of spatial relationships and constraints
- Reduces clutter in ER diagram and relational schema



Point Line Polygon

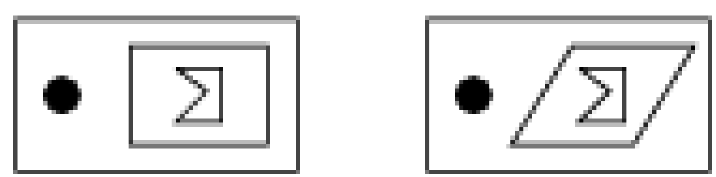
Pictograms for Basic Shapes



Pictograms Multishapes (using cardinality)

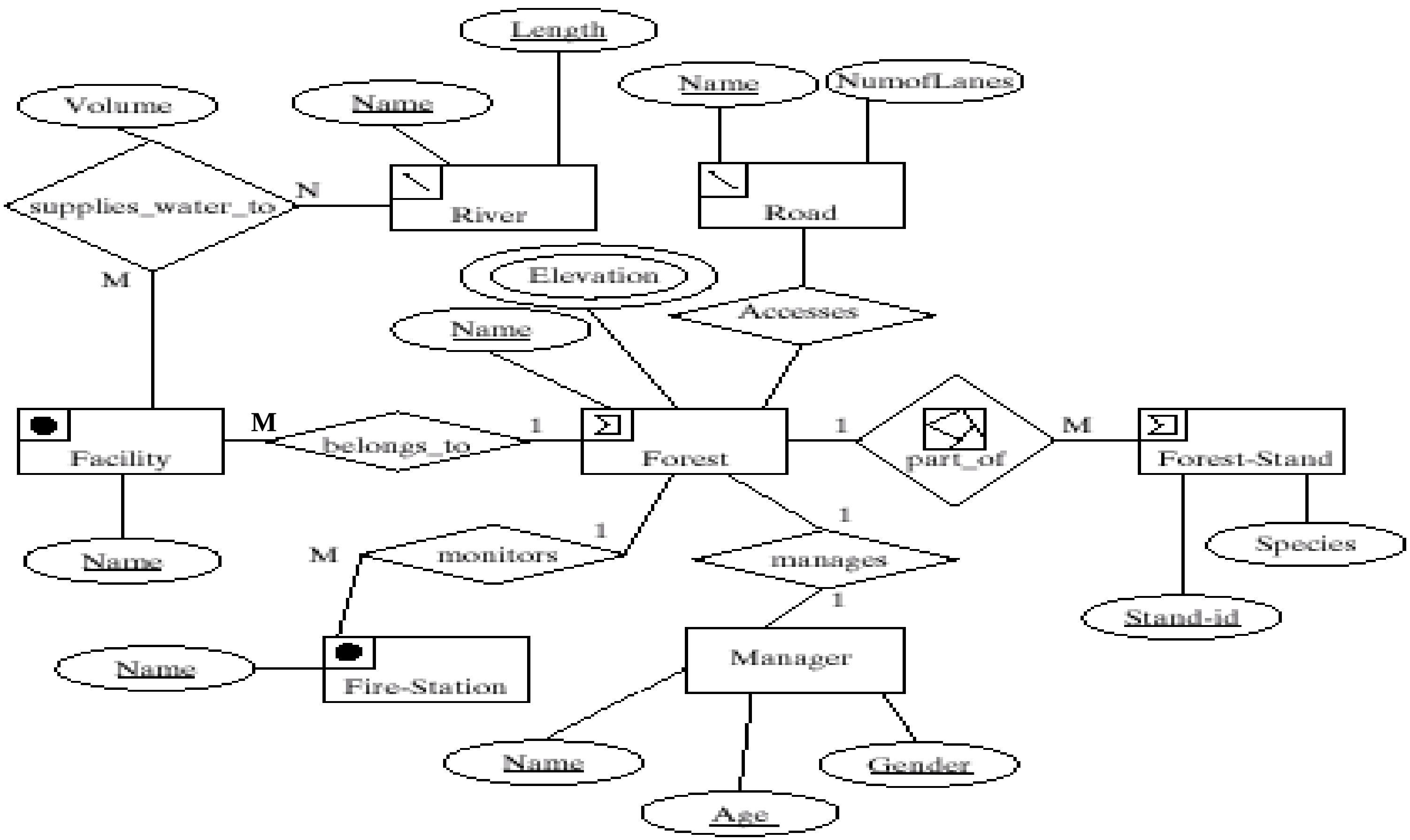
Part_of(Network) Part_of(Partition)

Pictograms for Relationships



Pictograms for Alternate Shapes

ER Diagram with Pictograms: An Example



For simplicity no
multishape pictograms

2.5 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

Summary

- 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