90-706

Lecture 4

Object Modeling

Richard J. Orgass
H. John Heinz III School of Public Policy and Management
Carnegie Mellon University

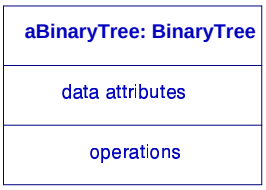
Agenda

- Objects and Classes
- Links and Associations

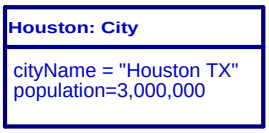
Definitions

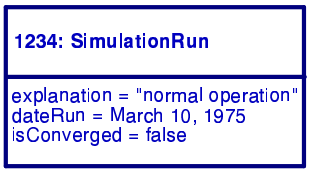
- An object is a concept, abstraction, or thing that has meaning for an application.
- Each object exists and can be identified.
- Identity is "that property of an object which distinguishes each object from all others". [Khoshafian-86]
- A class is a set of objects with similar properties, common behavior (operations and state diagrams), similar relationships to other objects and common semantics.

Object Diagrams

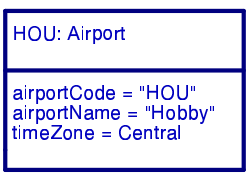


Examples of Objects

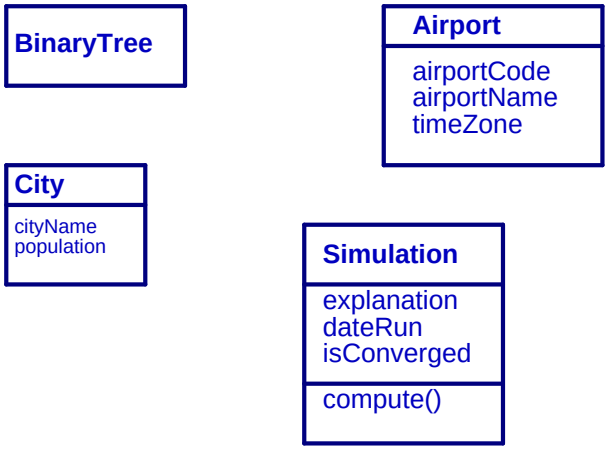




More Examples of Objects

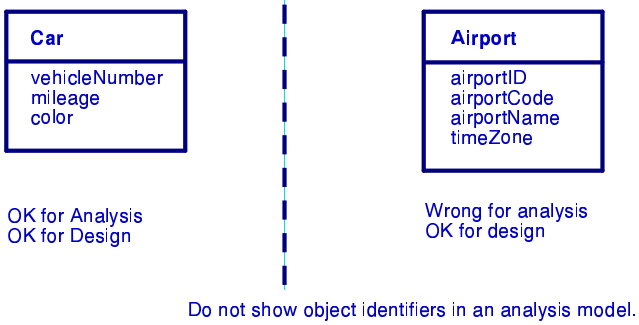


Class Diagrams



Values and Objects

- During design, you may show internal identifiers. For example, you may use internal identifiers to clarify the design of relational database tables.

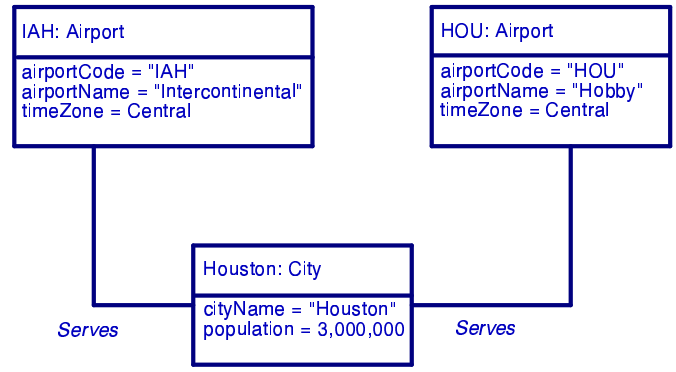


Do not show object identifiers in an analysis model.

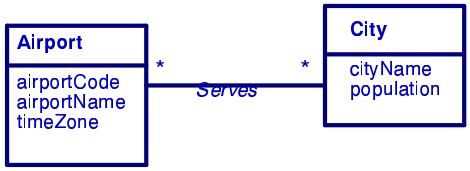
Links and Associations

- A **link** is a physical or conceptual connection between objects.
  - most relate two objects
  - some relate three or more objects
  - is not a value
- An **association** is a description of a group of links with common structure and common semantics.
- A link is an instance of an association.
- The links of an association relate objects from the same classes and have similar properties (link attributes).

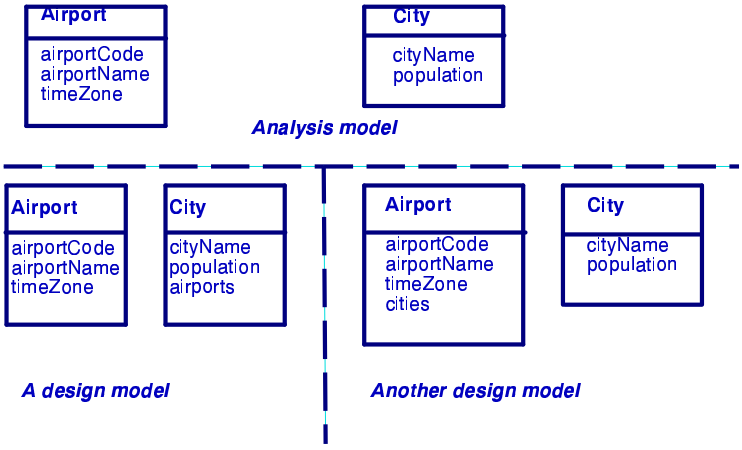
Links



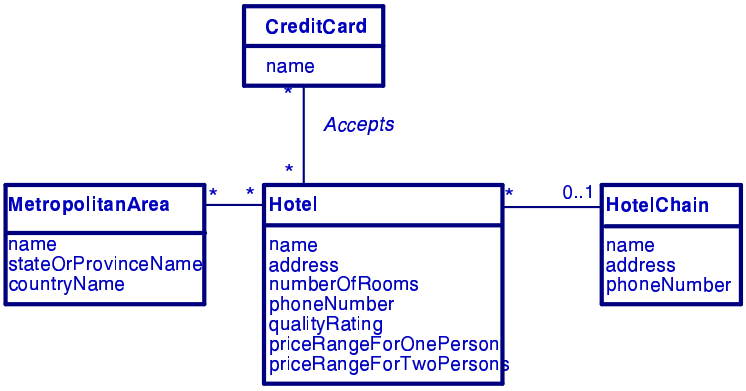
Associations



Associations
Analysis and Design Models



Associations in an Object Model

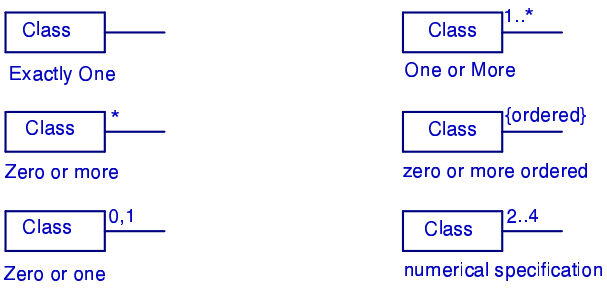


Associations in an object model for hotel selection.

Multiplicity

- **Multiplicity** specifies the number of instances of one class that may relate to a single instance of an associated class.
- Constraint on the size of a collection not a count of the members.

Multiplicity - 2



Labeling multiplicity of classes in object models.

In Class Exercise

- Add the following to the hotel selection model
  - A hotel may have a number of amenities, such as restaurant, lounge, fitness center, indoor pool, outdoor pool, golf and tennis. Extend the model so that you can easily obtain the set of amenities for a given hotel and the set of hotels that offer one or more amenities.
  - Many hotels have variable pricing throughout the year; rates are low during off seasons and high during seasons of high demand. Extend the model so that it organizes pricing information by season.

Common Associations List

- A is a physical part of B
  - Drawer -- POST
  - Wing -- Airplane
- A is a logical part of B
  - SalesLineItem -- Sale
  - FlightLeg -- FlightRoute
- A is physically contained in/on B
  - POST -- Store, Item -- Shelf
  - Passenger -- Airplane
- A is logically contained in B
  - ItemDescription -- Catalog
  - Flight -- FlightSchedule
- A is a description for B
  - ItemDescription -- Item
  - FlightDescription -- Flight

Common Associations List -- 2

- A is a line item of a transaction or report B
  - SalesLineItem -- Sale
  - MaintenanceJob -- MaintenanceLog
- A is known/logged/recorded/captured in B
  - Sale -- POST
  - Reservation -- FlightManifest
- A is a member of B
  - Cashier -- Store
  - Pilot -- Airline
- A is an organizational subunit of B
  - Department -- Store
  - Maintenance -- Airline
- A uses or manages B
  - Cashier -- POST
  - Pilot -- Airplane

Common Associations List -- 3

- A communicates with B
  - Customer -- Cashier
  - ReservationAgent -- Passenger
- A is related to a transaction B
  - Customer -- Payment
  - Passenger -- Ticket
- A is a transaction related to another transaction B
  - Payment -- Sale
  - Reservation -- Cancellation
- A is next to B
  - POST -- POST
  - City -- City
- A is owned by B
  - POST -- Store
  - Plane -- Airline

Association Guidelines

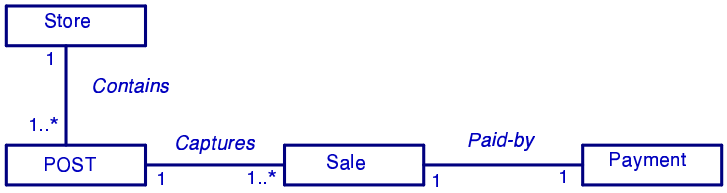
- How detailed should associations be?
  - Common pitfall in creating conceptual models is to spend too much time during investigation trying to discover them.
  - Finding concepts is much more important than finding associations. The majority of time spent in conceptual model creation should be devoted to indentifying concepts, not associations.
- Focus on those associations for which knowledge of the relationship needs to be preserved for some duration (need-to-know) associations).
- It is more important to identify concepts than to identify associations.
- Too many associations tend to confuse a conceptual model rather than illuminate it. Their discovery can be time consuming, with marginal benefit.
- Avoid showing redundant or derivable associations.

Roles

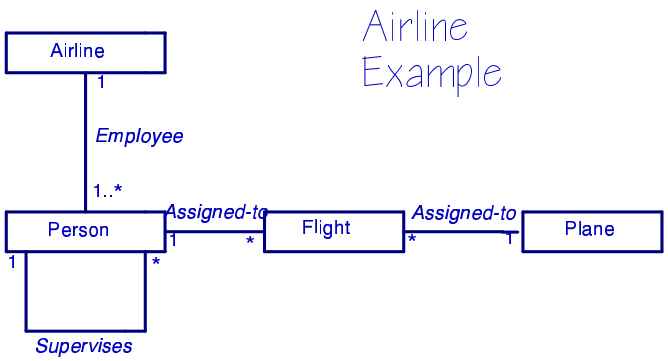
- Each end of an association is called a role.
- Roles may optionally have:
  - name
  - multiplicity expression
  - navigability

Naming Associations

- Name an association based on a **ClassName-VerbPhrase-ClassName** format where the verb phrase creates a sequence that is readable and meaningful in the model context.
- Store Example



Association Names

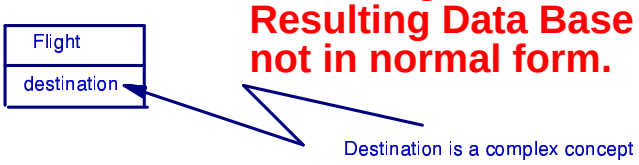


Relate with Associations not Attributes

Better

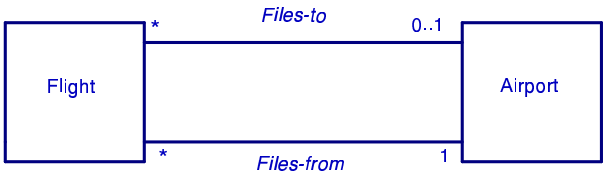
Complex Domains as Associations

Bad design
Resulting Data Base is not in normal form.



Better

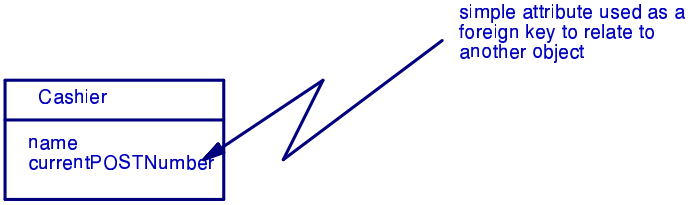
Multiple Associations Between Classes



(Not every flight is guaranteed to land at an airport!)

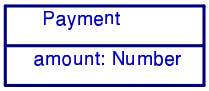
Design Creep: No Attributes as Foreign Keys

- Attributes should not be used to relate concepts in conceptual models or models created for database systems.
- **DO NOT** add foreign key attributes to associate two types -- use an association instead.

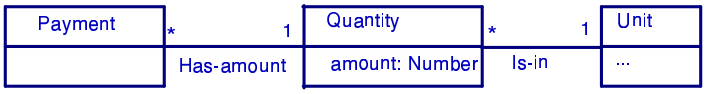


POOR DESIGN
Use a Foreign Key to relate to another object.

Modeling Quantities

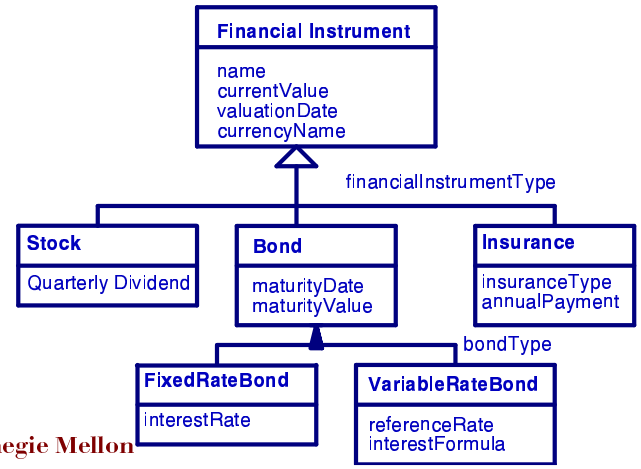


Usable, but not flexible or robust



Better

Generalization



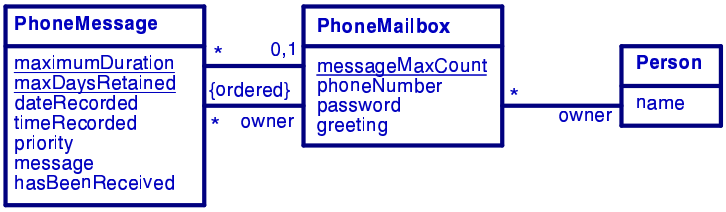
Attributes and Operations

- Class attributes
  - one copy for shared by all instances of a class
  - Example: **static** attributes of Java Classes
  - Underline names in class diagrams
- Class operations
  - operation on a class rather than on a member of the class (an object)
  - Example: **new** operation of Java
- Class Instance
  - An object that is a member of a class
  - Example: the return value of the operation **new** in Java

Avoid Class Attributes

- Model Groups Explicitly
- Using class attributes imposes restrictions
  - All class instances must have the same value for these attributes
  - Prevents adding different restrictions later

Model Groups Explicitly



POOR DESIGN

All users have same
maximumDuration
maxDaysRetained
messageMaxCount

Can't change these for individual users or for periods of time