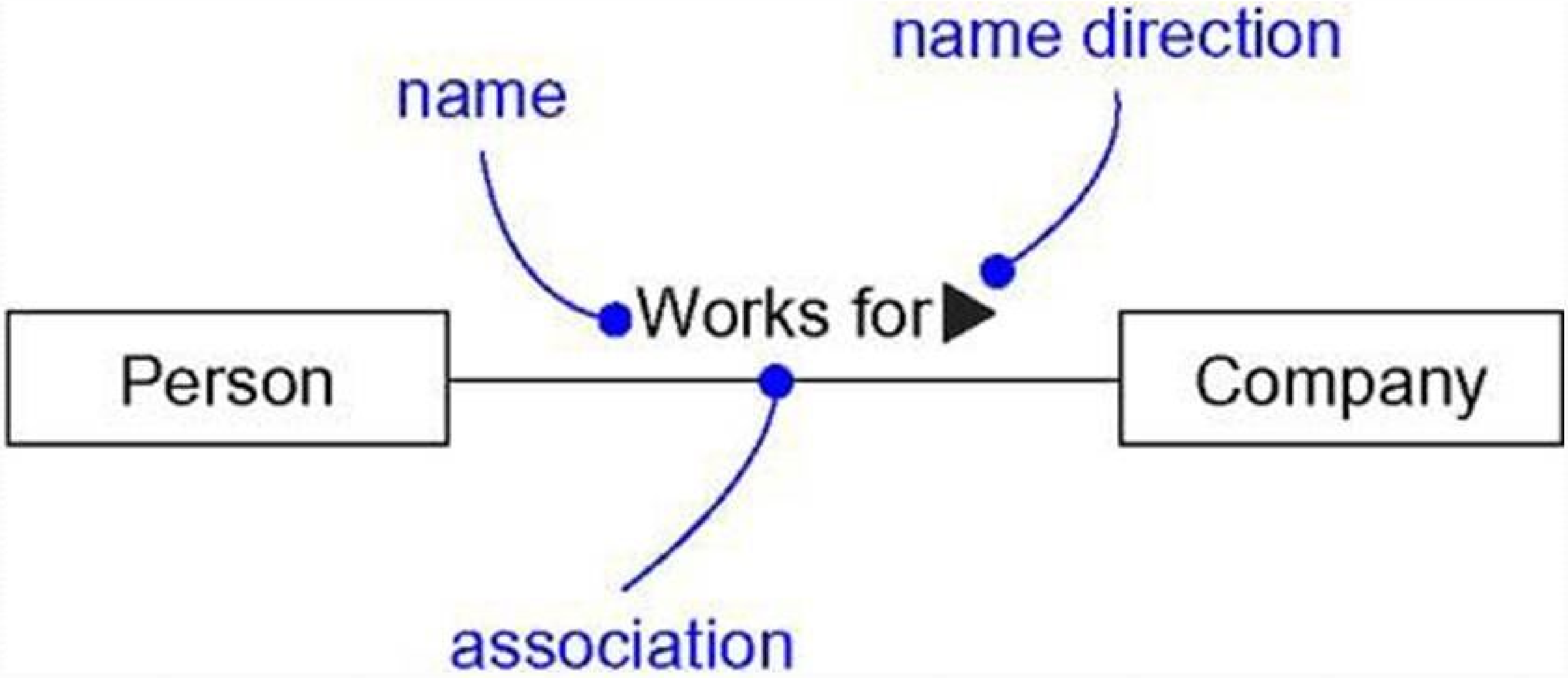
UML (Class Diagrams)

A Conceptual Model of the UML

- Relationships:
  - Dependency
  - Association
  - Generalization
  - Realization.

- An association is …
  - a type of relationship that shows a 'knows-a' relationship.
  - either unidirectional or bidirectional.
  - represented by a solid line which may optionally be labeled and have a name direction indicator or navigability arrows.
  - an alternative notation for a class *attribute*
  - Association names are verbs or verb phrases.
  - The same class can be on both ends of an association.

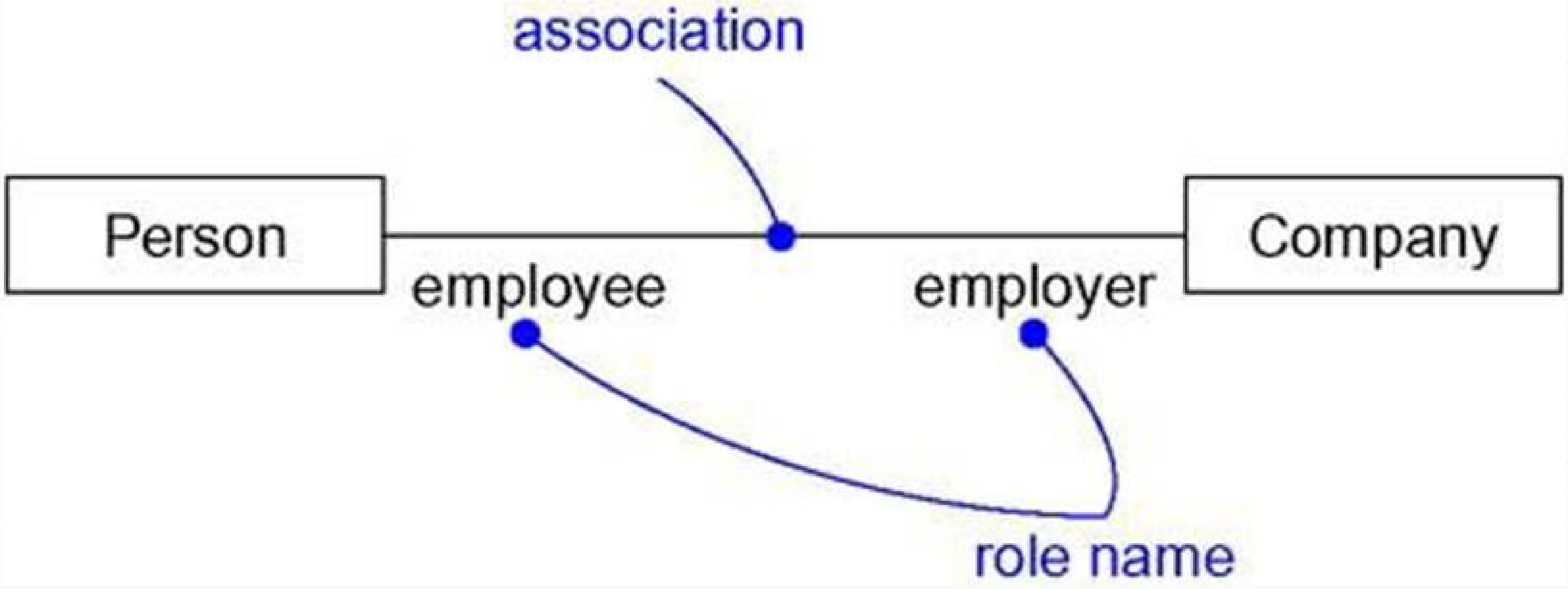
Binary Association



Association name versus role name

Associations

- When a class participates in an association, it has a specific role that it plays in that relationship
- Associations may optionally have role names on either end of the association

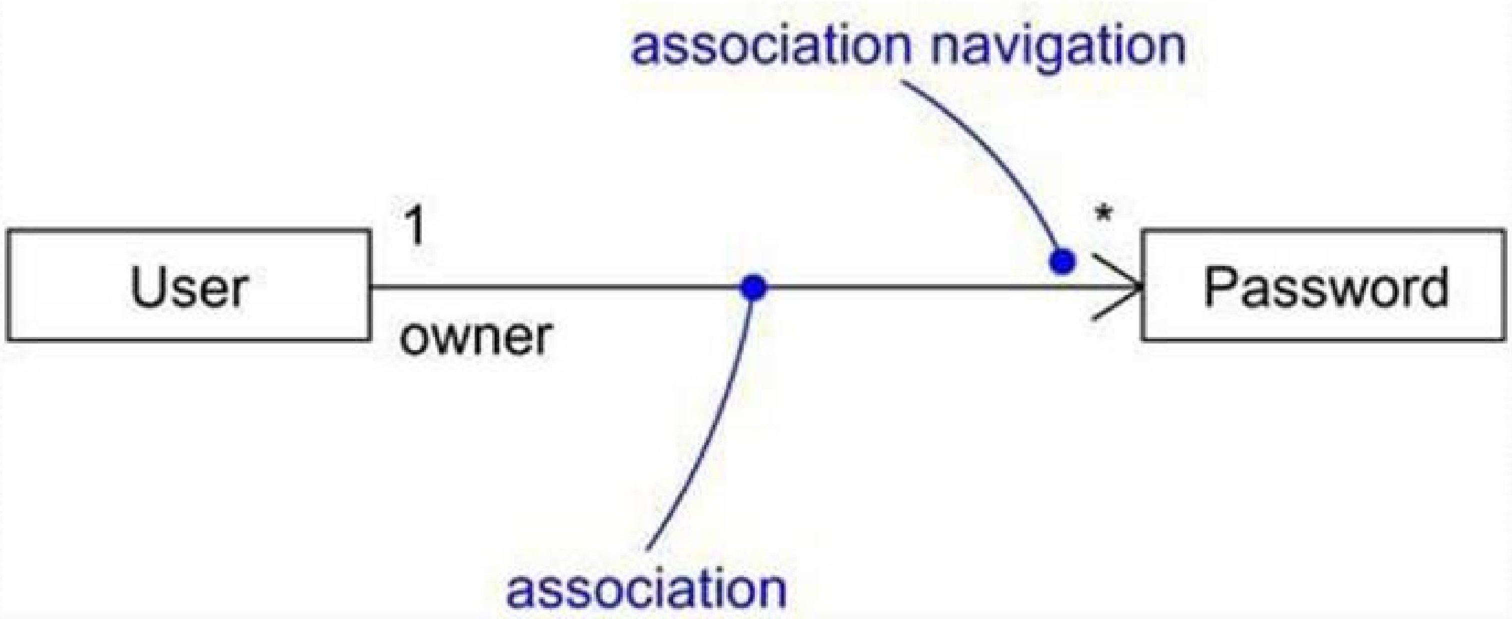


Associations

- Role
  - A class can participate in many associations and thus have multiple (different) roles.
  - The face that a class on one end of an association presents to the class on the other end of the association.
  - Role names are nouns.
  - Role names are usually used in place of association names.

Navigation

- Association between two classes: person and Company
- Navigation across an association is bidirectional by default
- However, there are some circumstances in which you'll want to limit navigation to just one direction.
- For example, an association between User and Password objects.
- Direction of navigation

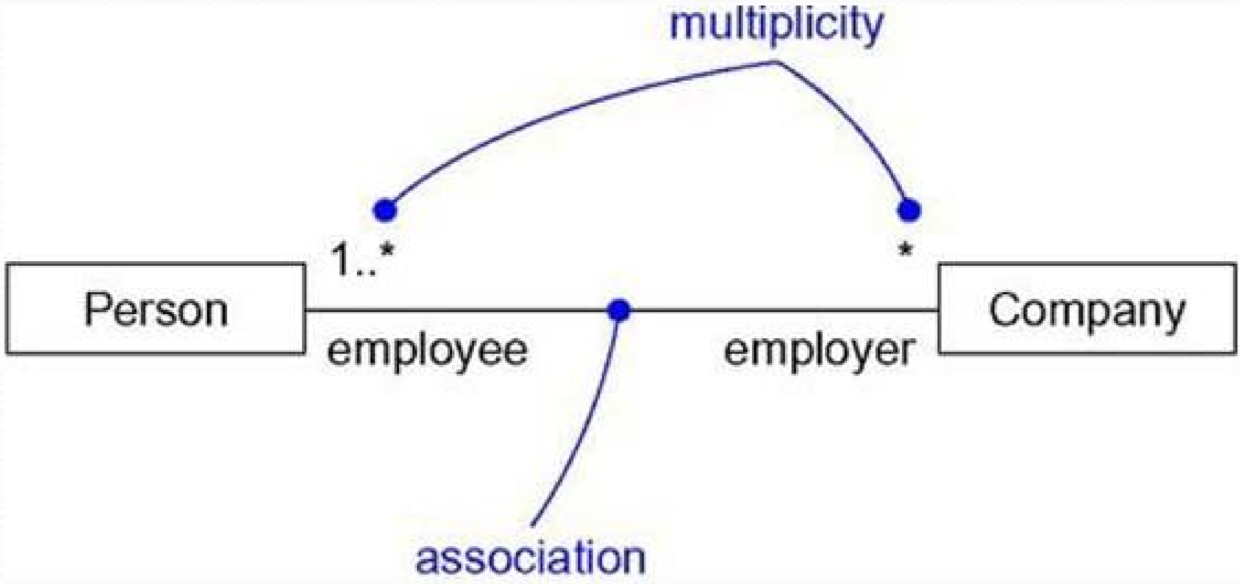


Navigation

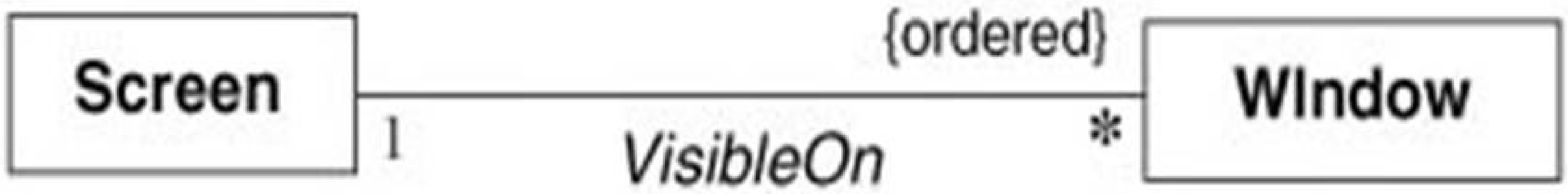
- Specifying that an association is navigable means that, given an object at one end, you can easily get to objects at the other end
- Usually because the source object stores some references to objects of the target.
- However this does not mean that we can never traverse in other direction
- Association is often implemented as a reference attribute in one object that refers to another object

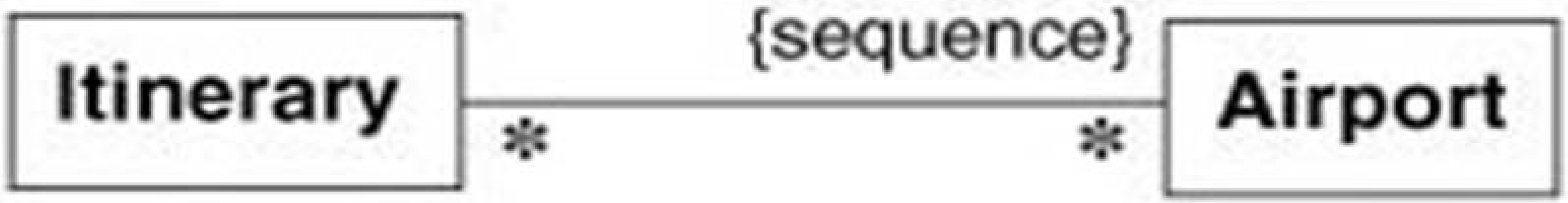
Multiplicity

- Indicates how many object may be connected across an instance of an association.
- Commonly used multiplicities
  - 1 (default)
  - 0..1
  - * (or 0..*)
  - [***lower-bound*** '..'] ***upper-bound***
  - 0..1, 3..4, 6..*



- Elements in a multi valued multiplicity form a set
- Order and uniqueness of the collection elements.

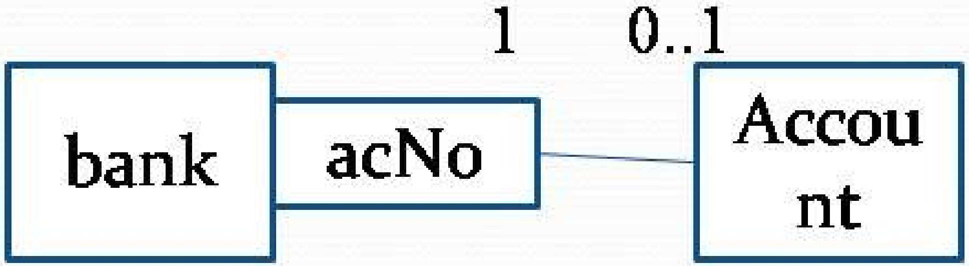


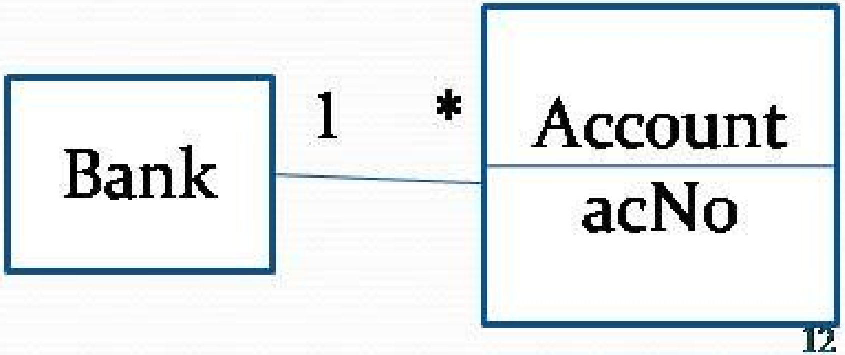


Collection Type	isOrdered	isUnique
Multiset, bag	false	false
Sequence, list	true	false
Set	false	true
Ordered set	true	true

Qualified Associations

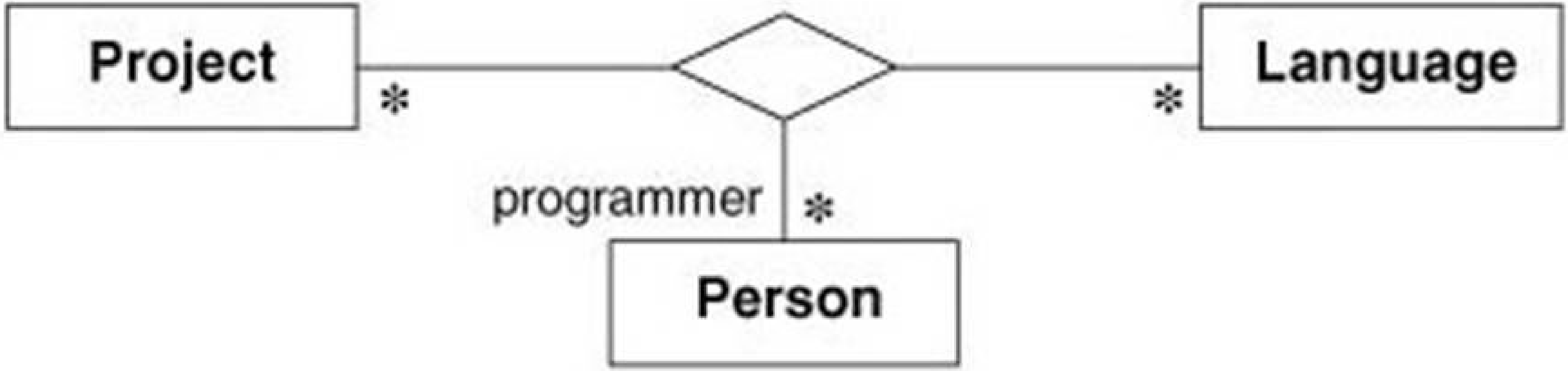
- Qualified model adds information
- Increases the precision of a model
- An attribute called the qualifier selects from the target objects reducing the effective multiplicity from many to one
- All access to a given account require an acNo as an argument **suggesting an implementation using a key and value data structure.**



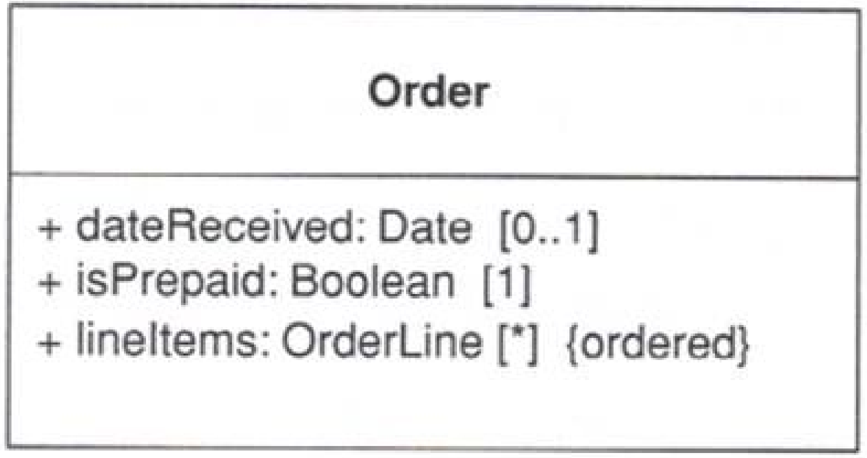


N-ary associations

Programmers use computer languages on projects



Attributes Modeled as Associations



Order

+ dateReceived: Date [0..1]
+ isPrepaid: Boolean [1]
+ lineItems: OrderLine [*] {ordered}

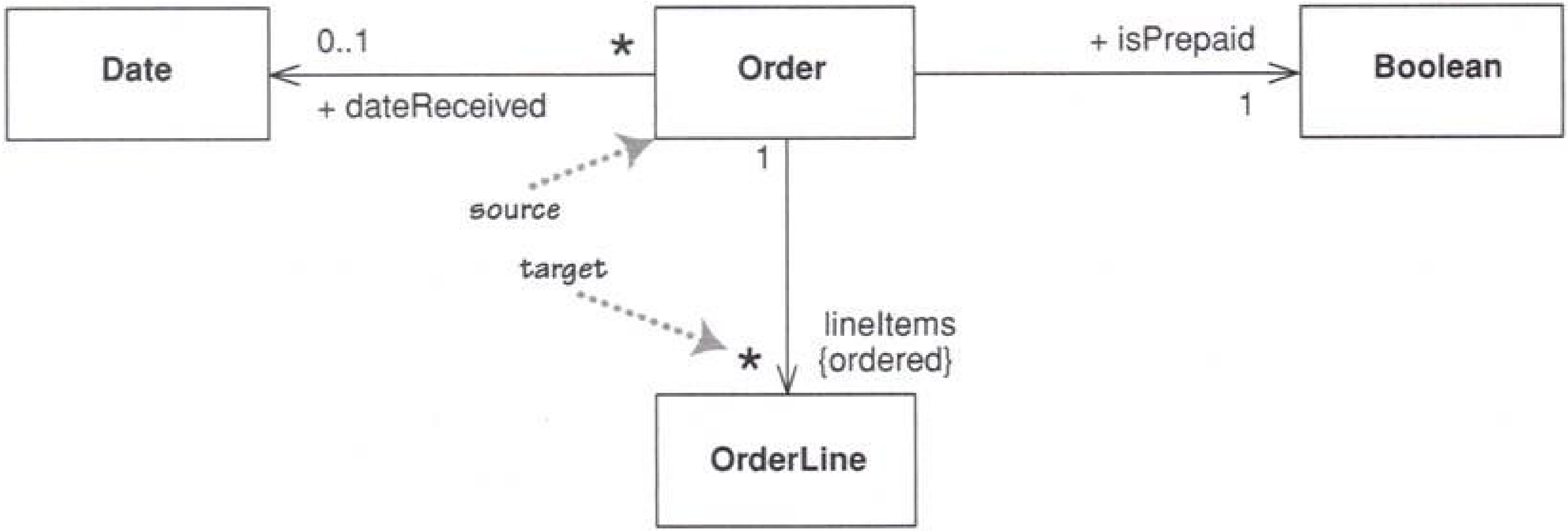
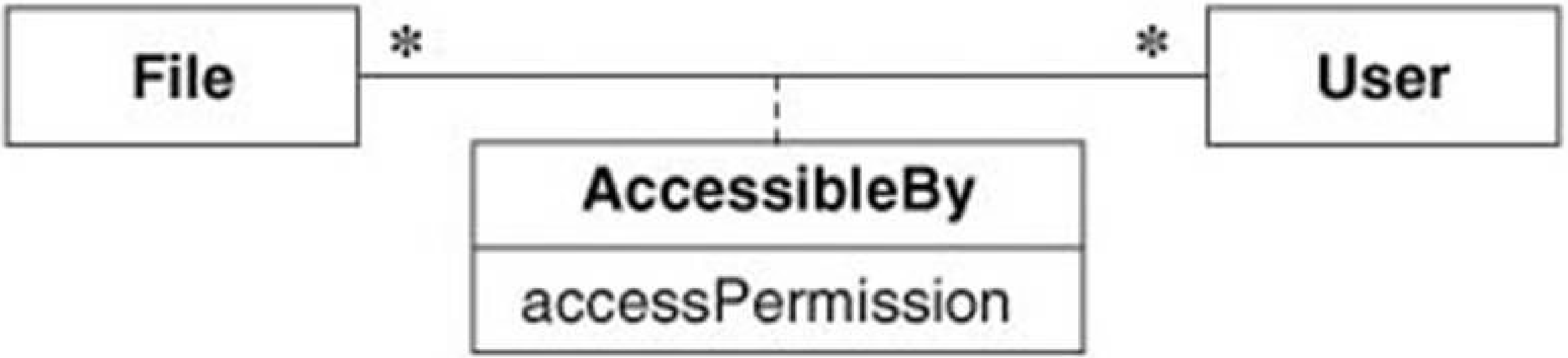


Figure 3.3 *Showing properties of an order as associations*

Association Class

- A File (class) is accessible by (association) a user (class).
- Association has property access permission.

-> represent association as association class.



/etc/termcap	read	John Doe
/etc/termcap	read-write	Mary Brown
/usr/doe/.login	read-write	John Doe

Association Class

- An *association class allows to add attributes, operations and other features to an association*
- *Shown by a class symbol attached by a dashed line to an association*
- You can't attach the same class to more than one association; an association class is the association
- There can be only one instance of the association class between any two participating objects
- The name of the association is usually omitted since it is considered to be the same as that of the attached class.

Association Class

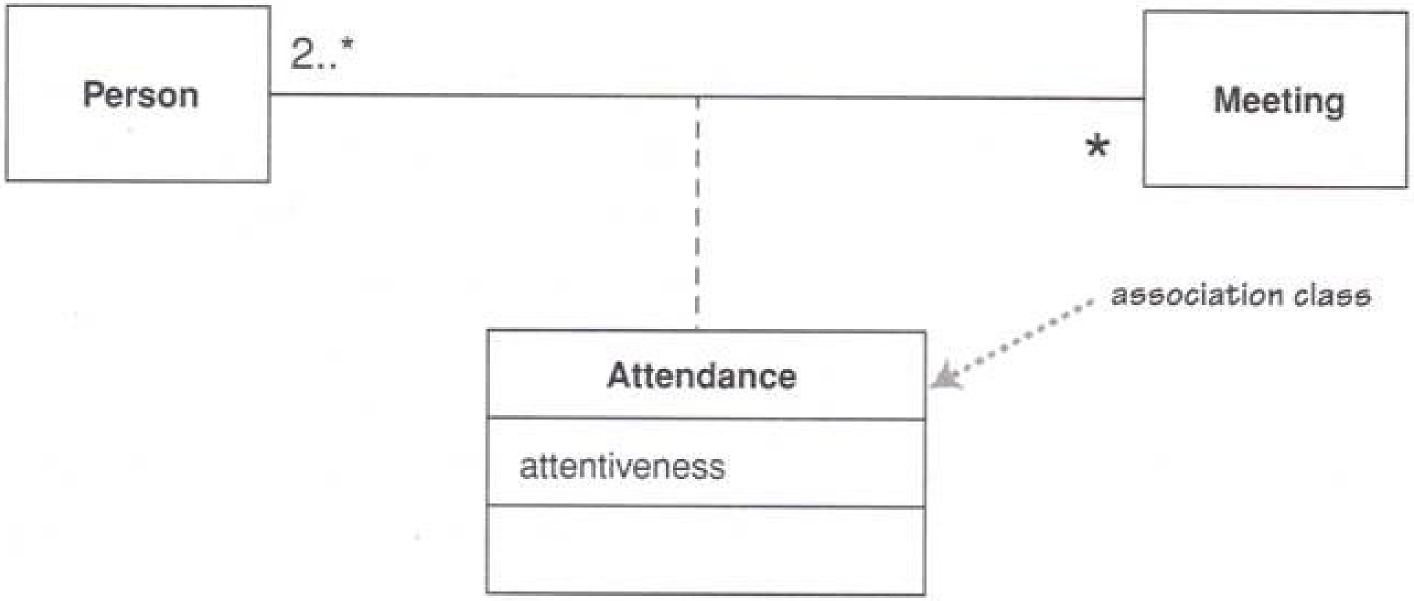


Figure 5.12 *Association class*

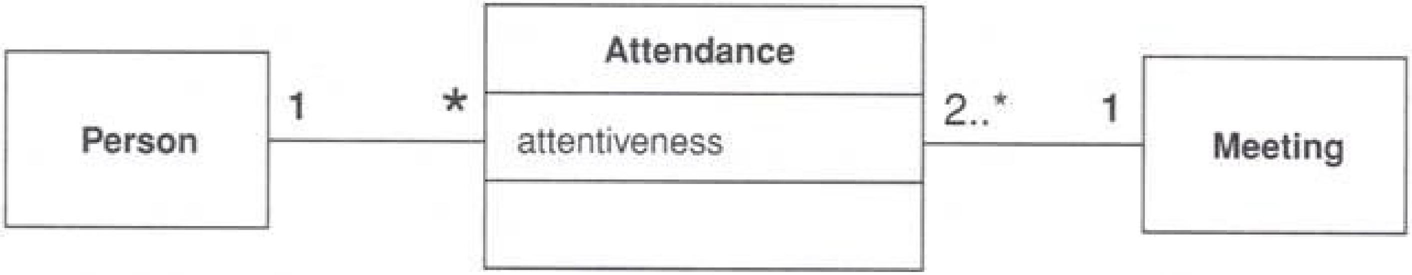


Figure 5.13 *Promoting an association class to a full class*

Association Class

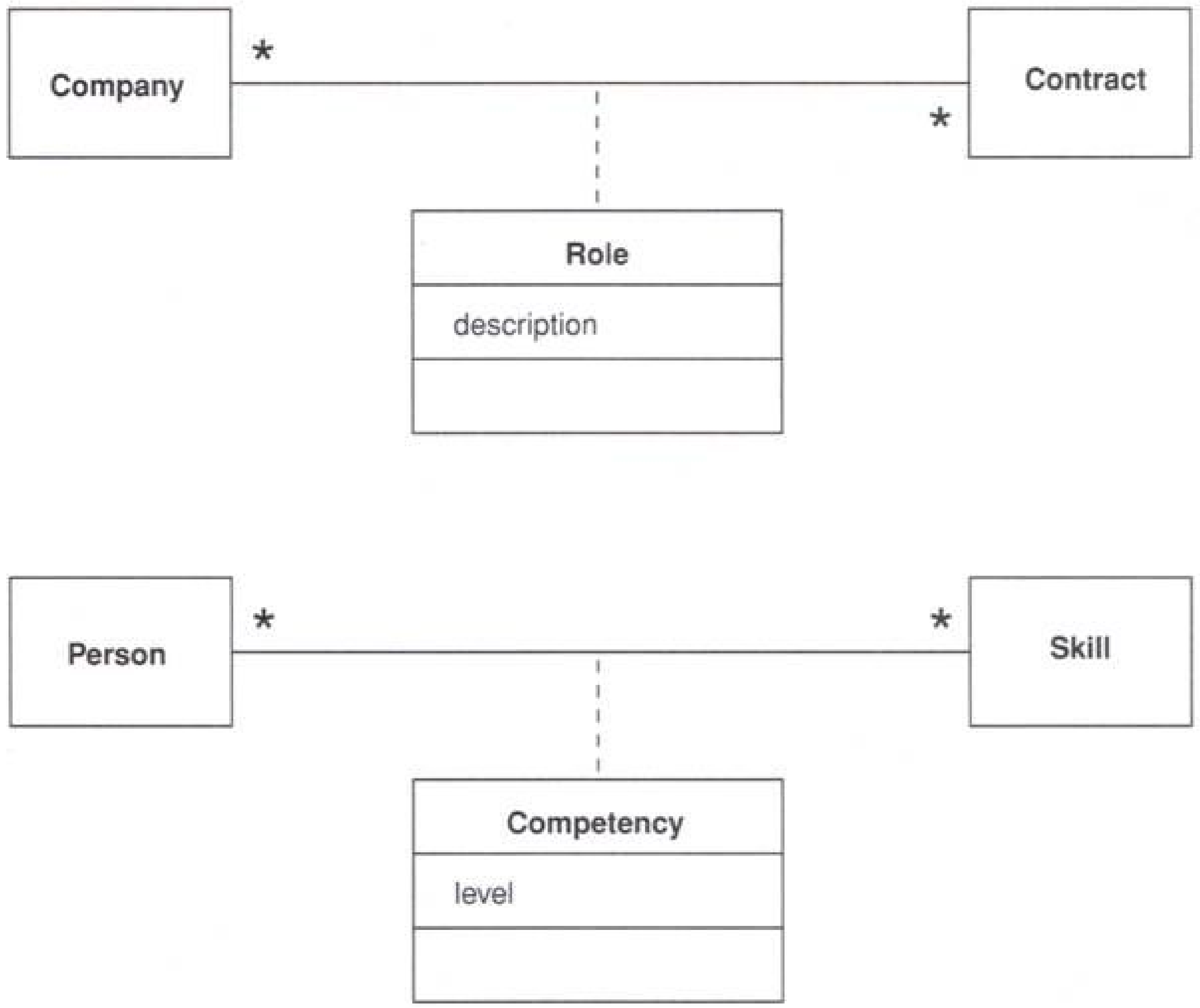


Figure 5.14 *Association class subtleties (Role should probably not be an association class)*