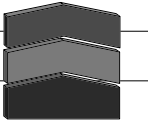


Associations & Attributes



• **Have concepts (?)**

- For conceptual model

• **Now need to add**

- Associations between concepts
- Attributes of concepts

---

---

---

---

---

---

---

---

Association Definition

• **Larman**

- “a relationship between concepts that indicates some meaningful and interesting connection”

• **UML (v1.1)**

- “semantic relationship between two or more classifiers that involves connections among their instances”

---

---

---

---

---

---

---

---

Knowledge of a Relationship

• **Preserved for some duration**

• **Links between concepts (objects)**

- SalesLineItem with Sale ?
- Sale with Manager ?

• **Association types**

- “Need-to-know”
- “Comprehension-only”

---

---

---

---

---

---

---

---

## Finding Associations



### • Common associations (pp. 108-109)

- Whole/part (physical or logical)
- Ownership
- Communication

### • Conceptual model guideline

- Concepts are most important
- Associations are secondary
  - Many will be discovered later

---

---

---

---

---

---

---

---

## Roles



### • Each end of an association

- Represents role that object plays

### • Role has:

- Name (clarifies association name?)
- Multiplicity
- Navigability
- Other details?

---

---

---

---

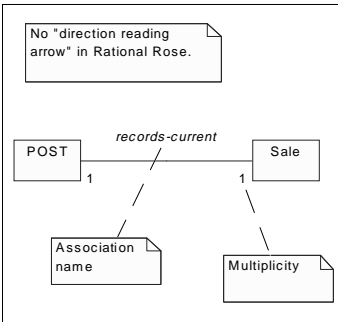
---

---

---

---

## UML Association Notation



Compare Figure 10.2, page 107.

Multiplicity:  
 \* : any number  
 1..\* : one/more  
 1..3 : range  
 4 : exact #

How do we count?

---

---

---

---

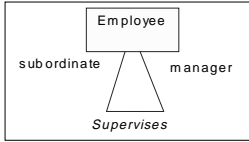
---

---

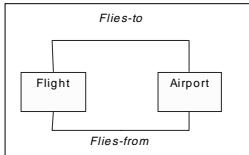
---

---

### Special Cases of Associations



Reflexive association



Multiple associations between classes/concepts

---

---

---

---

---

---

---

---

### Attribute Definition

- **Larman**
  - “a logical data value of an object”
- **UML (v1.1)**
  - “a named slot within a classifier that describes a range of values that instances of the classifier may hold”

---

---

---

---

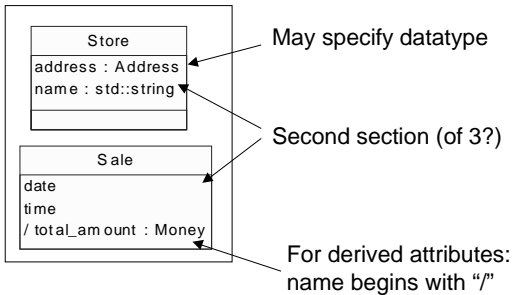
---

---

---

---

### UML Attribute Notation




---

---

---

---

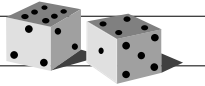
---

---

---

---

### Choosing Attributes



- **Keep them simple**
  - Boolean, date, number, string
- **Don't encode relationships**
  - No foreign keys! (database term)
- **Pure data values**
  - Not object instances with identity
- **When in doubt, make it a concept**

---

---

---

---

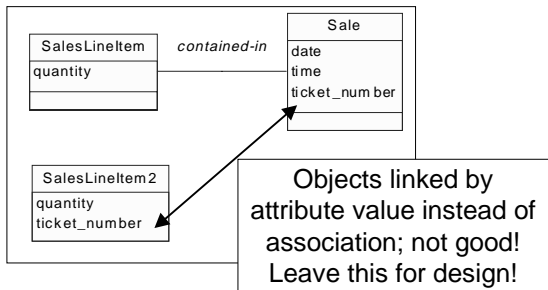
---

---

---

---

### Foreign Keys




---

---

---

---

---

---

---

---

### Non-Primitive Types



- **Invent new type for attribute?**
- **Examples:**
  - Name (just a string?)
  - Social Security Number (integer?)
- **Create non-primitive type**
  - If segmented, has associated operations, has other attributes, has an associated unit, ... Separate concept?

---

---

---

---

---

---

---

---