

# Object Oriented Analysis and Design

## Conceptual Modelling

COMP3607  
Object Oriented Programming II

Week 3- 19 Sept-2018

1

## Terminology

Concept: real-world things

Operation: a service that can be requested from an object to effect behaviour

Class: software specifications and implementations

Method: implementation of an operation specifying the operation's algorithm or procedure

2

## Associations

An association is a relationship between concepts that indicates some meaningful and interesting connection.

Those worth noting usually imply knowledge of a relationship that needs to be preserved for some duration (ms, days, years) depending on the context.

Essentially, do we need to record or keep track of that relationship?

3

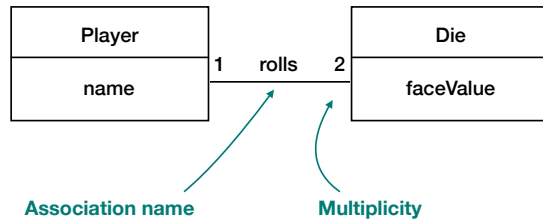
## Criteria for Useful Associations

In a conceptual model, the following associations should be considered and included:

- Associations for which knowledge of the relationship needs to be preserved for some duration (need-to-know associations)
- Associations derived from the Common Associations list

4

## Notation



5

## Finding Associations

Category	Example
★ A is a physical part of B	Drawer - POS
★ A is a logical part of B	SalesLineItem - Sale
★ A is physically contained in/on B	POS-Store
★ A is logically contained in B	ItemDescription - Catalogue
A is a description for B	ItemDescription - Item
A is a line item of a transaction or report of B	SalesLineItem - Sale
★ A is known/logged/recorded/captured in B	Sale - POS
A is a member of B	Cashier - Store
A is an organisational subunit of B	Department - Store

6

## Finding Associations

Category	Example
A uses or manages B	Cashier - POS
A communicates with B	Customer - Cashier
A is related to a transaction B	Customer - Payment
A is a transaction related to another transaction B	Payment - Sale
A is next to B	POS - POS
A is owned by B	POS - Store

★ High Priority

7

## Association Guidelines

- Focus on those associations for which knowledge of the relationship needs to be preserved for some duration.
- Identifying concepts is more important than identifying associations.
- Too many associations then to confuse a conceptual model rather than give insight.
- Avoid showing redundant or derivable associations

8

## Role

The end of an association is called a role.

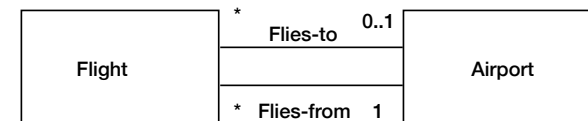
Roles may optionally have:

- Name
- Multiplicity expression: \*, 1..\*, 1..40, 5, or 3,4,5
- Navigability

9

## Multiple Associations

Two concepts may have multiple associations between them. If these are distinctly different relationships, then they should be shown separately.



10

## Attributes

An attribute is a logical data value of an object.

The attributes in a conceptual model should preferably be simple attributes or pure data values.

E.g. Boolean, Date, Number, String (text), Time

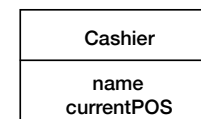
Other common types

Address, Colour, Geometrics, Phone number, Universal Product Code (UPC) etc.

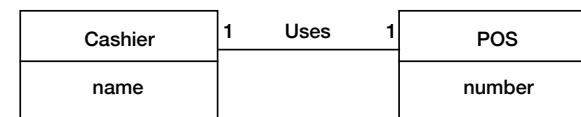
11

## Keep Attributes Simple

**Worse**

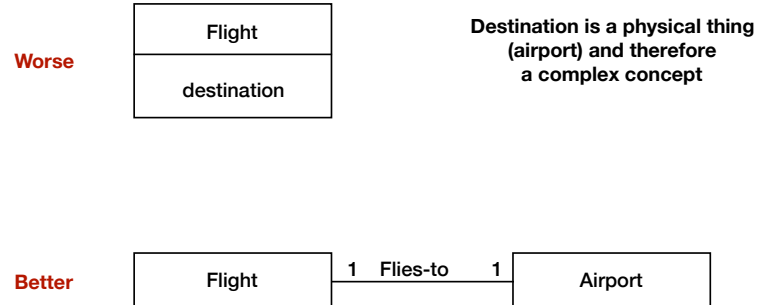


**Better**



12

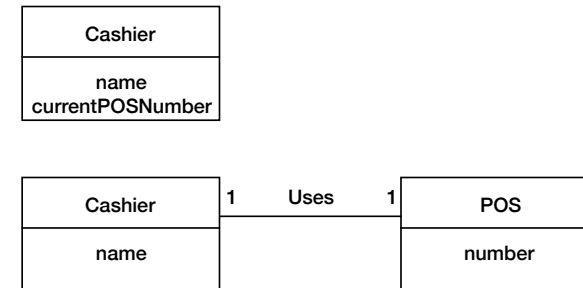
## Relate Concepts with Associations (not attributes)



13

## Design Creep

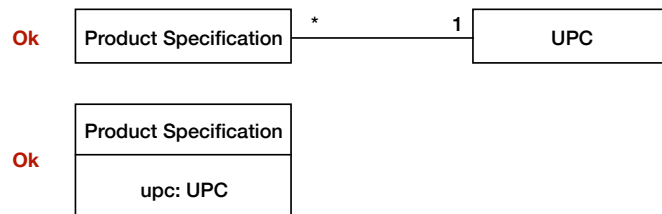
Attributes should not be used to relate concepts in the conceptual model. Most common violation of this principle is to add a kind of foreign key.



14

## Where to Illustrate? Non-primitive attributes and pure data values

Should a product code be shown as a separate concept in a conceptual model? It depends on what you want to emphasise in the diagram.



If the attribute is a pure data value, it may be shown in the attribute box

15

## How to Make a Conceptual Model

1. List the candidate concepts (using noun phrase identification, concept category list) related to the current requirements under consideration.
2. Draw them in a conceptual model
3. Add the associations necessary to record relationships for which there is some need to preserve memory
4. Add the attributes necessary to fulfil the information requirements

16

# System Sequence Diagrams

A system sequence diagram illustrates events from actors to systems. It shows for a particular scenario of a use case, the events that external actors generate, their order and intersystem events.

All systems are treated as a black box; the emphasis of the diagram is the events that cross the system boundary from actors to systems.

System behaviour describes what a system does without explaining how it does that.

17

# System Event

A system event is an external input event generated by an actor to a system. An event initiates a responding operation.

A system operation is an operation of the system that executes in response to a system event.

18

## Example: System Sequence Diagram

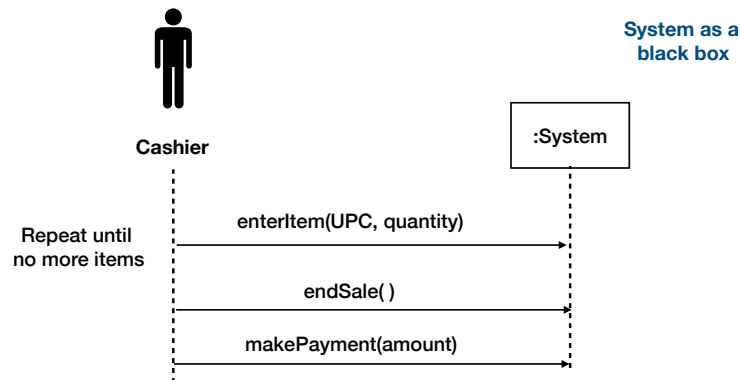


Figure: System Sequence Diagram for the Buy Items use case

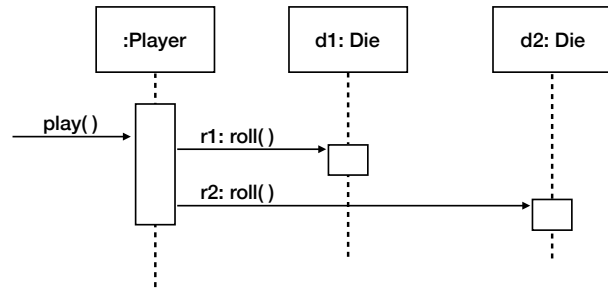
19

## How to Make a System Sequence Diagram

1. Draw a line representing the system as a black box
2. Identify each actor that directly operates on the system. Draw a line for each such actor
3. From the use case typical course of events text, identify the system (external) events that each actor generates. Illustrate them on the diagram
4. Optionally, include the use case text to the left of the diagram

20

## Example: Sequence Diagram



**Figure: Sequence Diagram of the Dice Game**  
Illustrating Message Flow and Method Invocation

21

## References

- UML: online reading resources
  - [www.uml.org](http://www.uml.org) (Documentation, notation)
  - [www.omg.org](http://www.omg.org) (Standards, protocols)
- Applying UML and Patterns, An Introduction to Object-Oriented Analysis and Design: Craig Larman. 1998.

22