

KSRA Learning Academy

KSRA Learning System

E-course

What Is Object-Orientation?



Copyright © 2020 Kavian Scientific Research Association

➤ REQUIREMENTS ANALYSIS

In This Lecture You Will Learn:

E-course

- ▶ Why we analyse requirements
- ▶ Technical terms used with class diagrams
- ▶ How the UML class diagram expresses a detailed model of user requirements
- ▶ How to realize use cases with collaboration diagrams and class diagrams
- ▶ How the CRC technique helps identify classes and allocate responsibilities



Why Analyse Requirements?

E-course

- ▶ Use case model alone is not enough
 - ▶ There may be repetition
 - ▶ Some parts may already exist as standard components
- ▶ Analysis aims to identify:
 - ▶ Common elements
 - ▶ Pre-existing elements
 - ▶ Interaction between different requirements

What a Requirements Model Must Do

E-course

A requirements model meets two main needs:

- ▶ Confirms what users want a new system to do
 - ▶ Must be understandable for users
 - ▶ Must be correct and complete
- ▶ Specifies what designers must design
 - ▶ Must be unambiguous



What a Requirements Model Must Do

E-course

- ▶ Describes what the software should do
- ▶ Represents people, things and concepts important to understand what is going on
- ▶ Shows connections and interactions among these people, things and concepts
- ▶ Shows the business situation in enough detail to evaluate possible designs
- ▶ Is organized so as to be useful for designing the software



How We Model the Analysis

E-course

- ▶ The main tool used for analysing requirements is the class diagram
- ▶ Two main ways to produce this:
 - ▶ Directly based on knowledge of the application domain (a Domain Model)
 - ▶ By producing a separate class diagram for each use case, then assembling them into a single model (an Analysis Class Model)

Class Diagram: Stereotypes

E-course

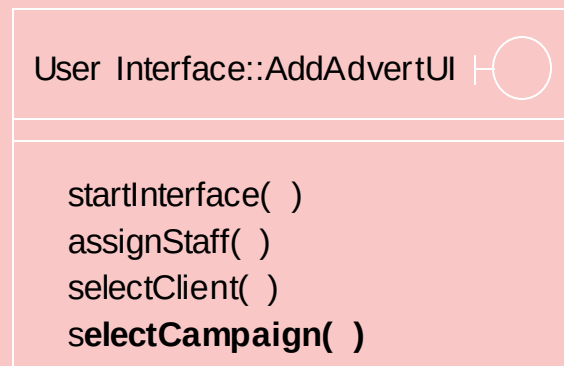
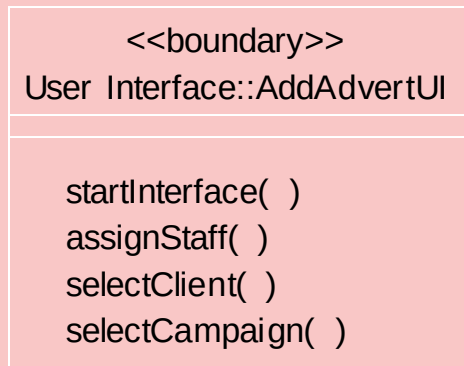
- ▶ Analysis class stereotypes differentiate the roles objects can play:
 - ▶ Boundary objects model interaction between the system and actors (and other systems)
 - ▶ Entity objects represent information and behaviour in the application domain
 - ▶ Control objects co-ordinate and control other objects



Class Diagram: Stereotypes

E-course

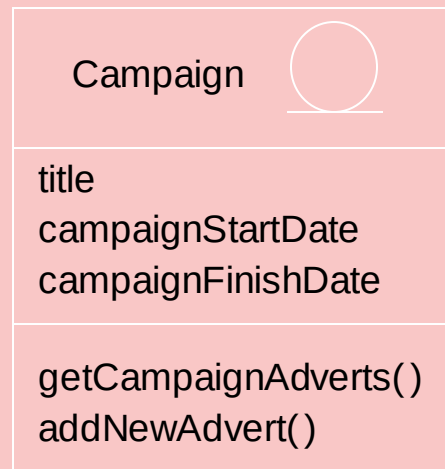
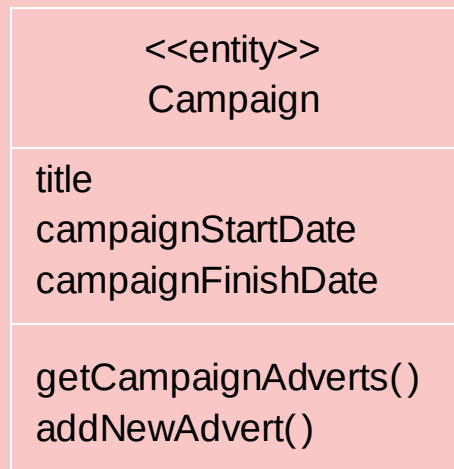
Alternative notations for boundary class:



Class Diagram: Stereotypes

E-course

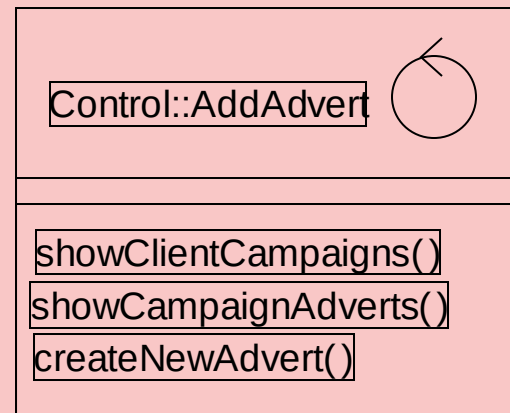
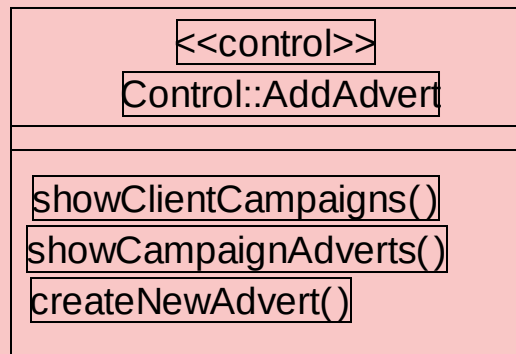
Alternative notations for entity class:



Class Diagram: Stereotypes

E-course

Alternative notations for control class:





Class Diagram: Instance Symbol

E-course

*Object name
compartment*

FoodCo:Client

Attribute values

companyAddress=Evans Farm, Norfolk
companyEmail=mail@foodco.com
companyFax=01589-008636
companyName=FoodCo
companyTelephone=01589-008638

*Instances do not
have operations*

Class Diagram: Attributes

E-course

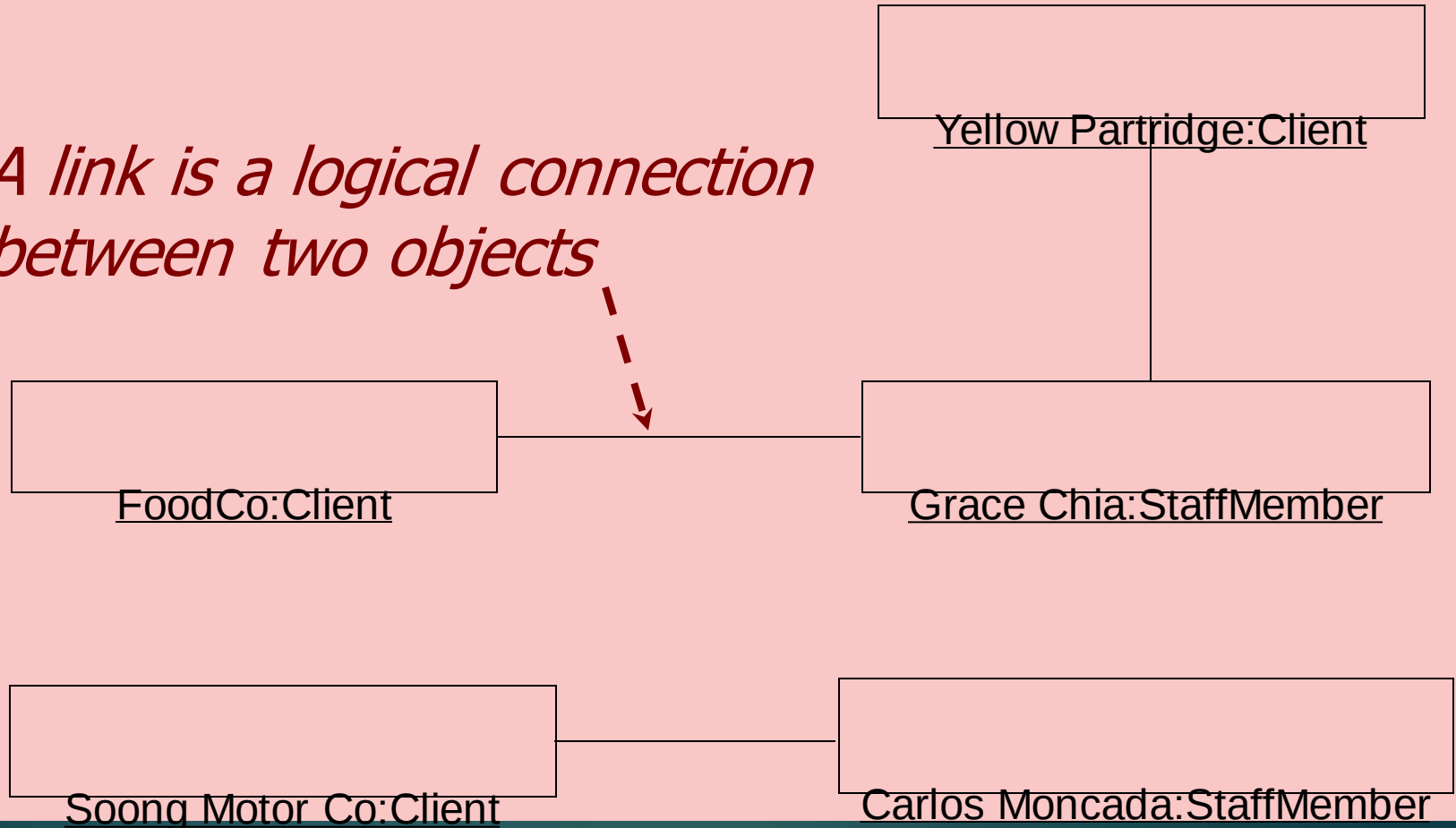
Attributes are:

- ▶ Part of the essential description of a class
- ▶ The common structure of what the class can ‘know’
- ▶ Each object has its own *value* for each attribute in its class

Class Diagram: Links

E-course

*A link is a logical connection
between two objects*



Class Diagram: Associations

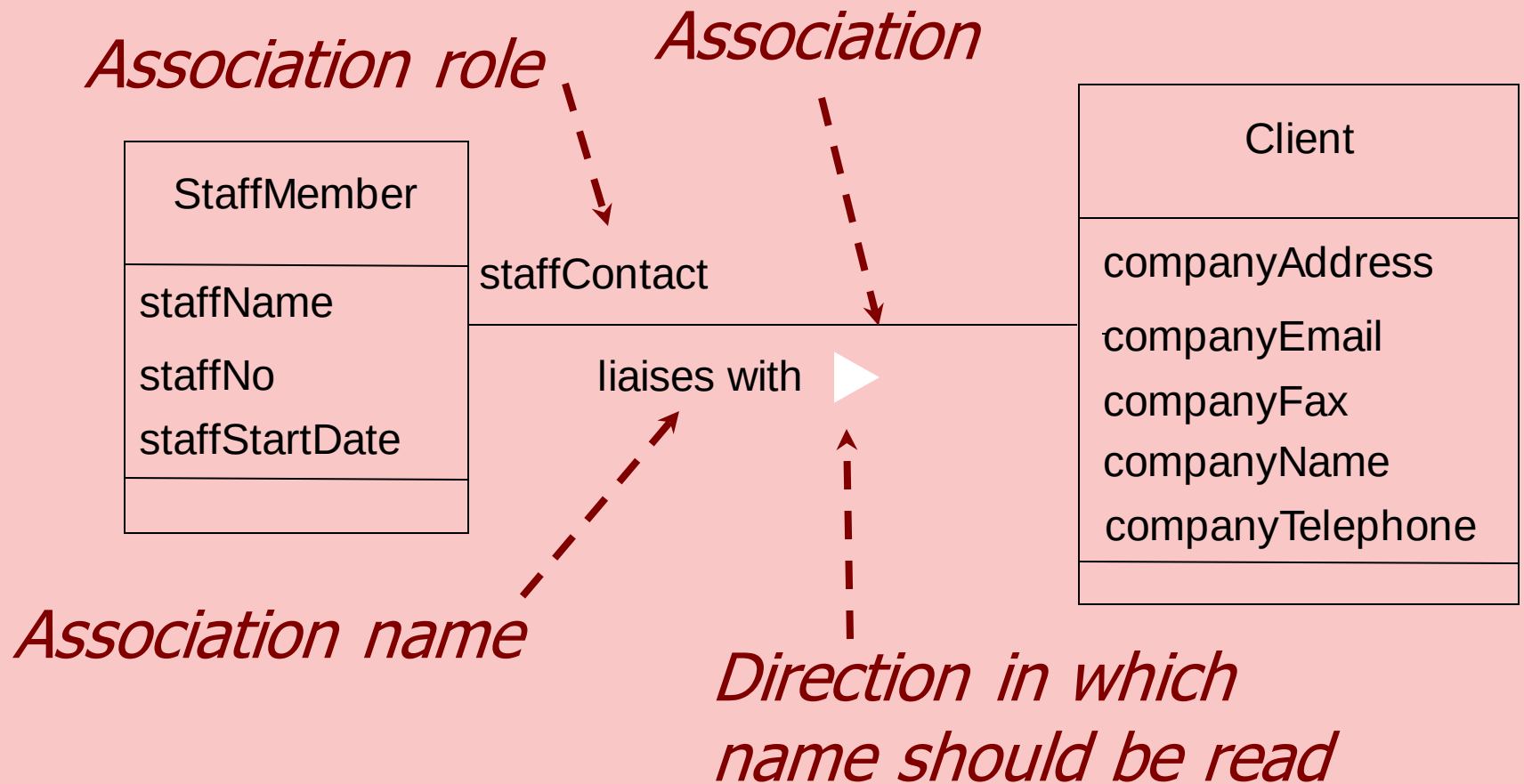
E-course

Associations represent:

- ▶ The possibility of a logical relationship or connection between objects of one class and objects of another
- ▶ If two objects can be linked, their classes have an association

Class Diagram: Associations

E-course



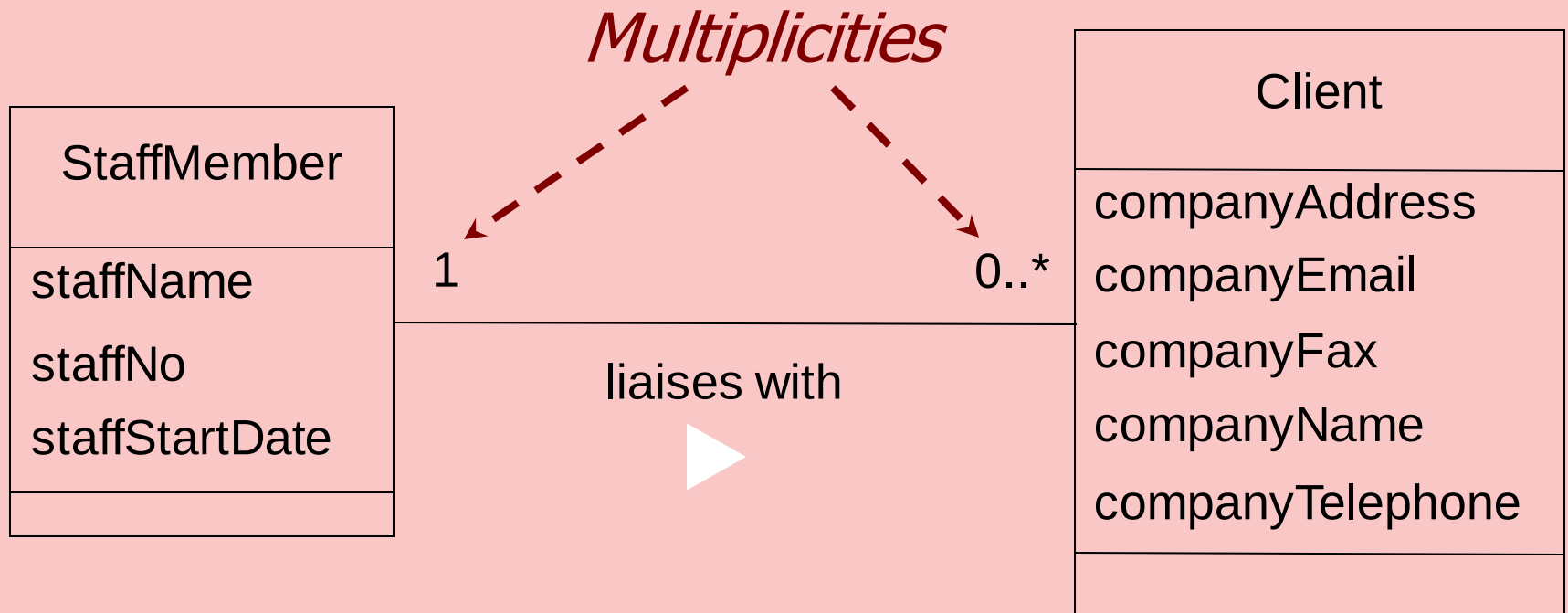
Class Diagram: Multiplicity

E-course

- ▶ Associations have multiplicity
- ▶ Multiplicity is the range of permitted cardinalities of an association
- ▶ Represent *enterprise* (or *business*) rules
- ▶ For example:
 - ▶ Any bank customer may have one or more accounts
 - ▶ Every account is for one, and only one, customer

Class Diagram: Multiplicity

E-course



- Exactly one staff member liaises with each client
- A staff member may liaise with zero, one or more clients

Class Diagram: Operations

E-course

Operations are:

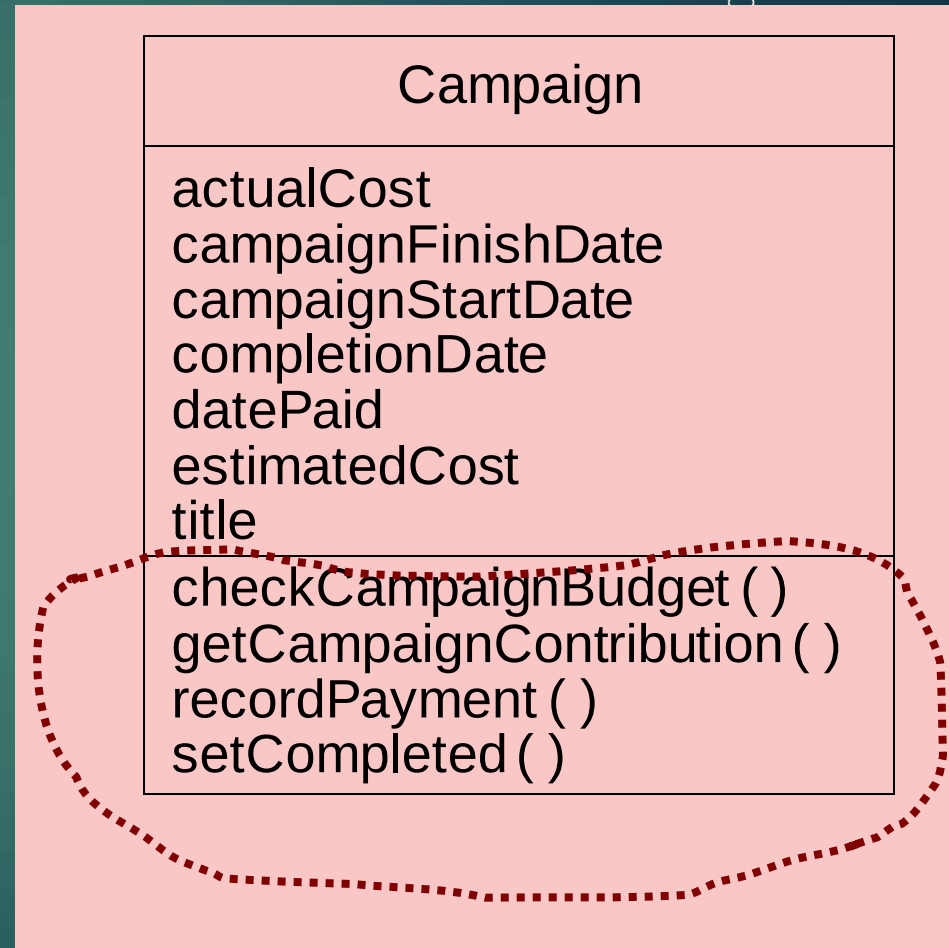
- ▶ An essential part of the description of a class
- ▶ The common behaviour shared by all objects of the class
- ▶ Services that objects of a class can provide to other objects

Class Diagram: Operations

E-course

010

- ▶ Operations describe what instances of a class can do:
 - ▶ Set or reveal attribute values
 - ▶ Perform calculations
 - ▶ Send messages to other objects
 - ▶ Create or destroy links



From Requirements to Classes

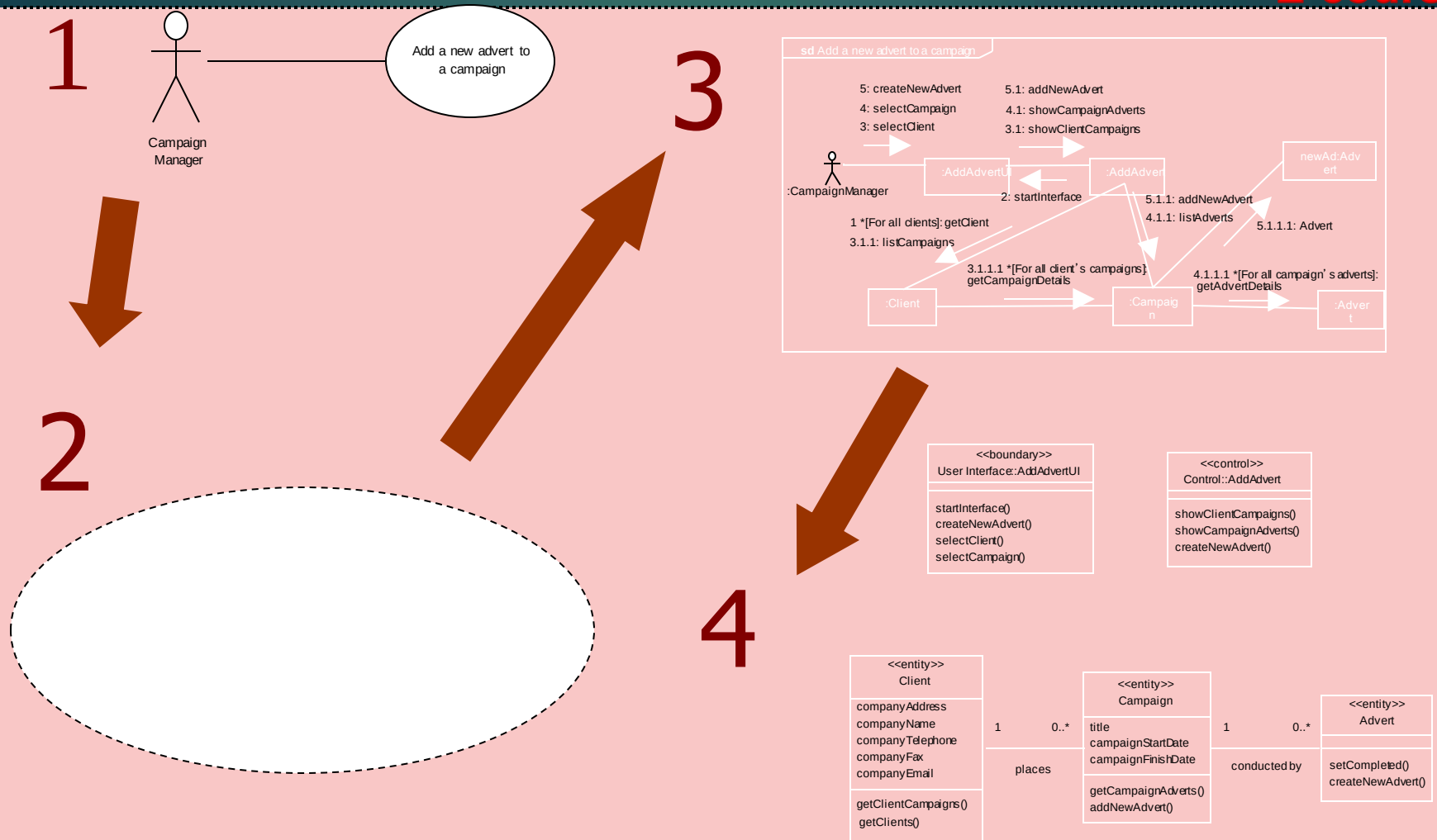
E-course

- ▶ Start with one use case
- ▶ Identify the likely classes involved (the use case collaboration)
- ▶ Draw a collaboration diagram that fulfils the needs of the use case
- ▶ Translate this collaboration into a class diagram
- ▶ Repeat for other use cases



From Requirements to Classes

E-course



Reasonability Checks for Candidate Classes (cont' d)

E-course

- ▶ A number of tests help to check whether a candidate class is reasonable
 - ▶ Is it beyond the scope of the system?
 - ▶ Does it refer to the system as a whole?
 - ▶ Does it duplicate another class?
 - ▶ Is it too vague?
 - ▶ (More on next slide)

Reasonability Checks for Candidate Classes (cont' d)

E-course

- ▶ Is it too tied up with physical inputs and outputs?
- ▶ Is it really an attribute?
- ▶ Is it really an operation?
- ▶ Is it really an association?
- ▶ If any answer is 'Yes', consider modelling the potential class in some other way (or do not model it at all)

CRC Cards

E-course

- ▶ Class–Responsibility–Collaboration cards help to model interaction between objects
- ▶ For a given scenario (or use case):
 - ▶ Brainstorm the objects
 - ▶ Allocate to team members
 - ▶ Role play the interaction

CRC Cards

E-course

Class Name:			
Responsibilities		Collaborations	
<i>Responsibilities of a class are listed in this section.</i>		<i>Collaborations with other classes are listed here, together with a brief description of the purpose of the collaboration.</i>	

Class Name <i>Client</i>	
Responsibilities	Collaborations
<i>Provide client information.</i> <i>Provide list of campaigns.</i>	<i>Campaign provides campaign details.</i>

Class Name <i>Campaign</i>	
Responsibilities	Collaborations
<i>Provide campaign information.</i> <i>Provide list of adverts.</i> <i>Add a new advert.</i>	<i>Advert provides advert details.</i> <i>Advert constructs new object.</i>

Class Name <i>Advert</i>	
Responsibilities	Collaborations
<i>Provide advert details.</i> <i>Construct adverts.</i>	

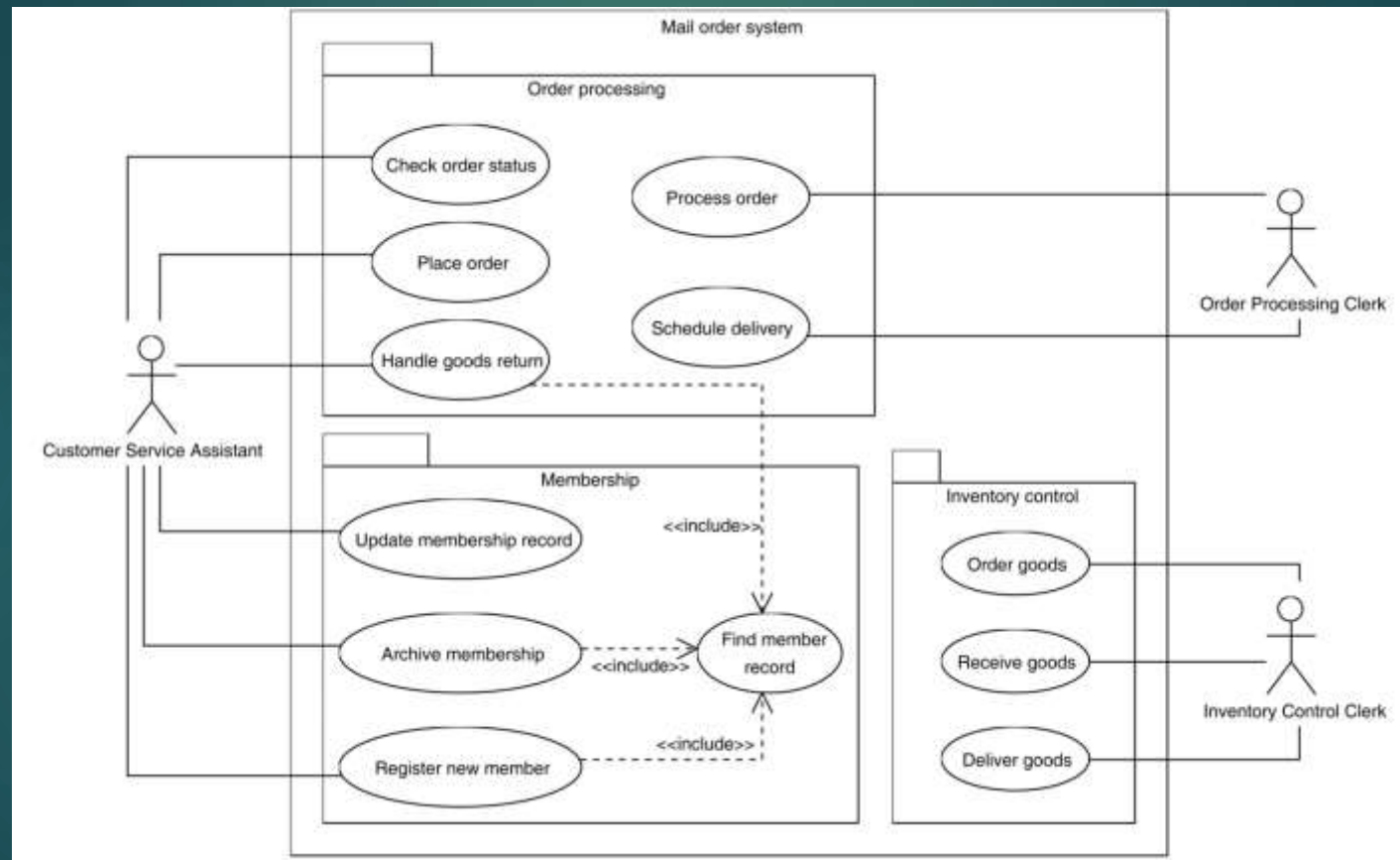
CRC Cards

E-course

- ▶ Effective role play depends on an explicit strategy for distributing responsibility among classes
- ▶ For example:
 - ▶ Each role player tries to be lazy
 - ▶ Persuades other players *their* class should accept responsibility for a given task
- ▶ May use 'Paper CASE' to document the associations and links

Requirements - Use Case Analysis: Structuring Use Case Model

E-course



Analysis

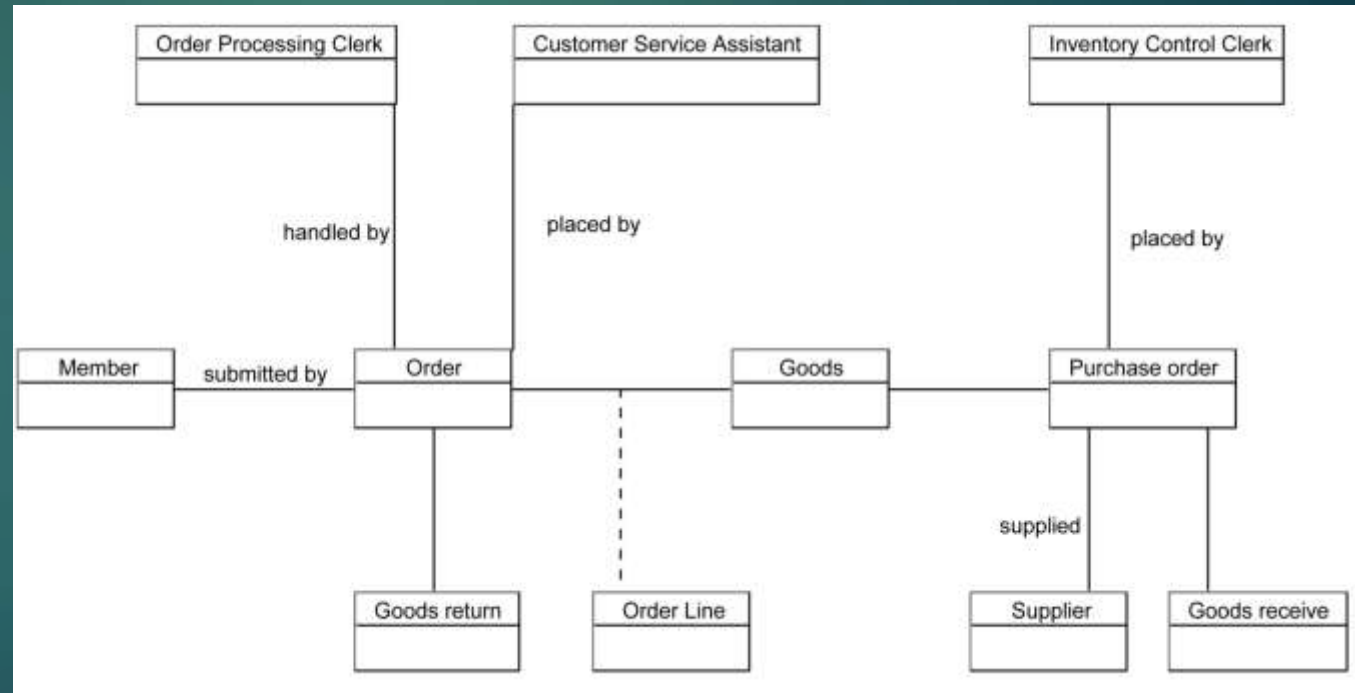
E-course

- ▶ The analysis workflow is to develop the domain class model and start the dynamic modeling.
- ▶ First, we develop the domain class model by applying the transitor(problem statement, domain class model) manipulator.
- ▶ Using this manipulator, we can construct the class model for the system (static modeling) and analyze the dynamic behaviors of the use cases (system modeling).

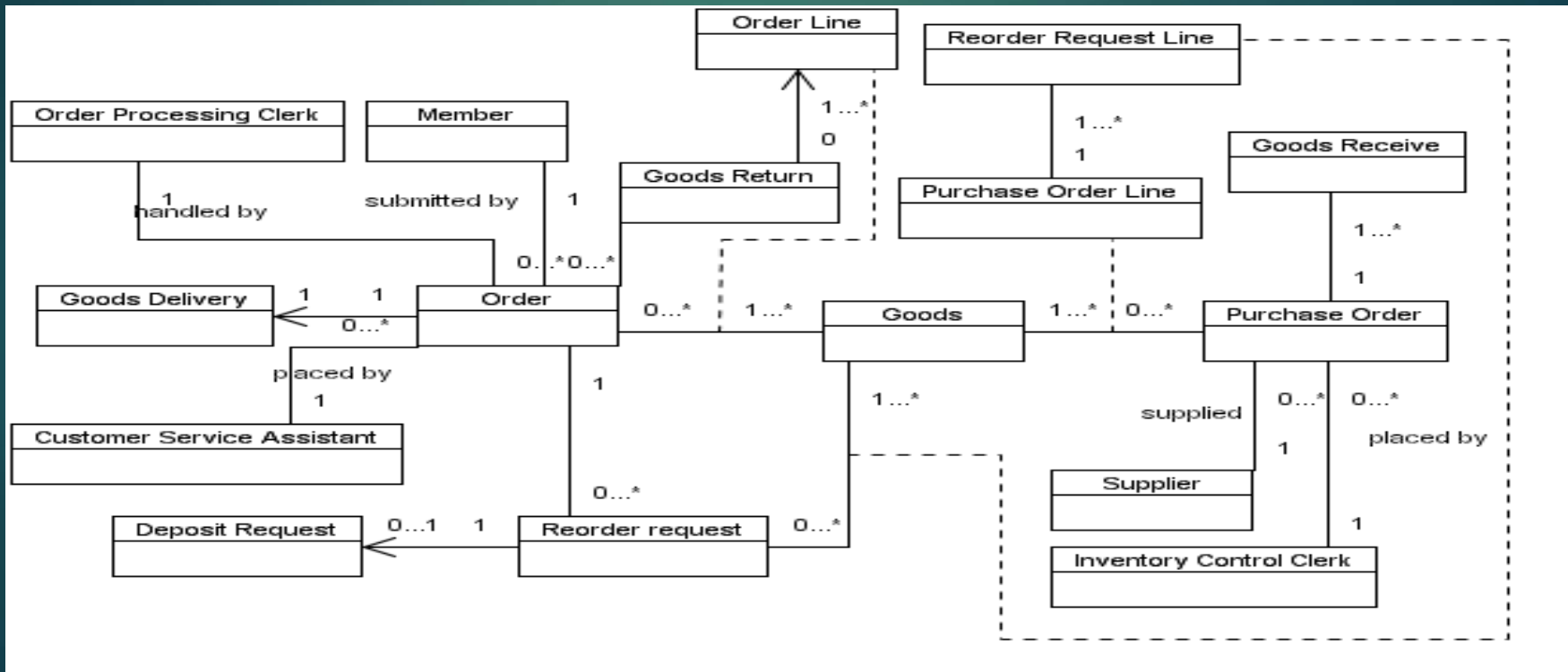
Analysis - Domain Analysis (Class Level)

E-course

- ▶ We apply the Transitor(Problem_Statement[use case level], Data_Dictionary.Class_List) manipulator to obtain the domain class model.



- Apply the Transitor (Use_Case_Description, Class_Diagram[analysis])



References

E-course

- ▶ Wirfs-Brock (1990) gives a good exposition of CRC cards
(For full bibliographic details, see Bennett, McRobb and Farmer)
- ▶ Object-Oriented Technology - From Diagram to Code with Visual Paradigm for UML, Curtis H.K. Tsang, Clarence S.W. Lau and Y.K. Leung, McGraw-Hill Education (Asia), 2005

WE FOCUS ON KNOWLEDGE-BASED ON EDUCATION

KSRA of Empowerment is a global non-profit organization committed to bringing empowerment through education by utilizing innovative mobile technology and educational research from experts and scientists. **KSRA** emerged in 2012 as a catalytic force to reach the hard to reach populations worldwide through Learning management system & E-learning & mobile learning.

The **KSRA** team partners with local under-served communities around the world to improve the access to and quality of knowledge based on education, amplify and augment learning programs where they exist, and create new opportunities for e-learning where traditional education systems are lacking or non-existent.

E-course

