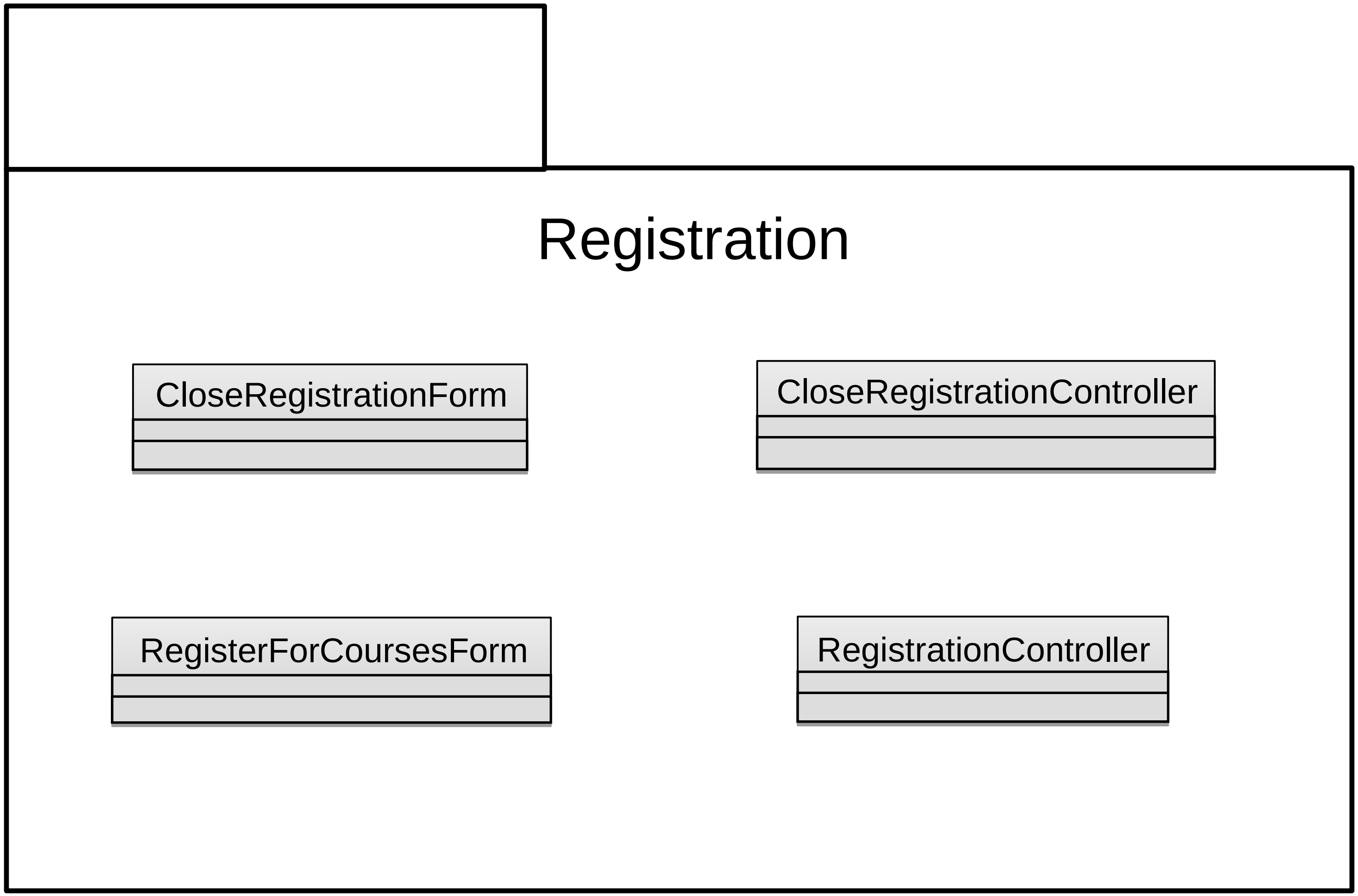
Conception Orientée Objets

Classes Structurées et Collaborations

Frédéric Mallet

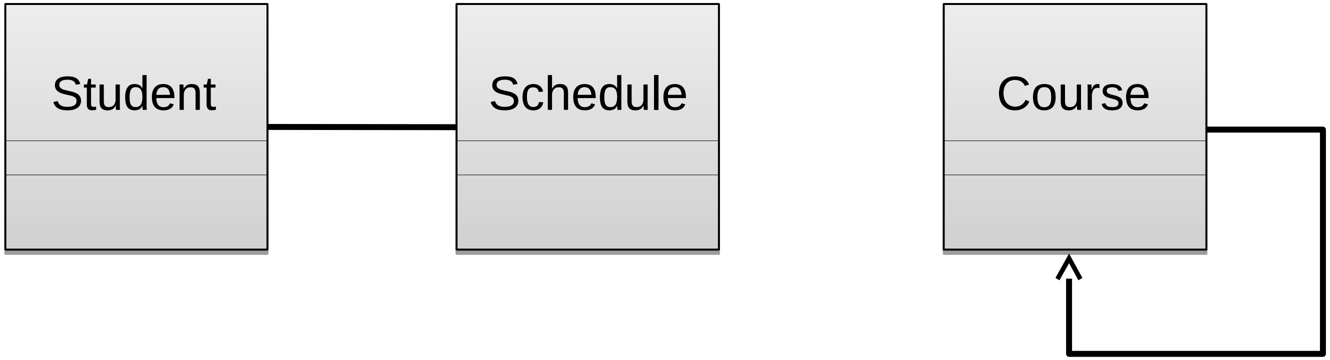
http://deptinfo.unice.fr/~fmallet/

Classes are grouped within Packages



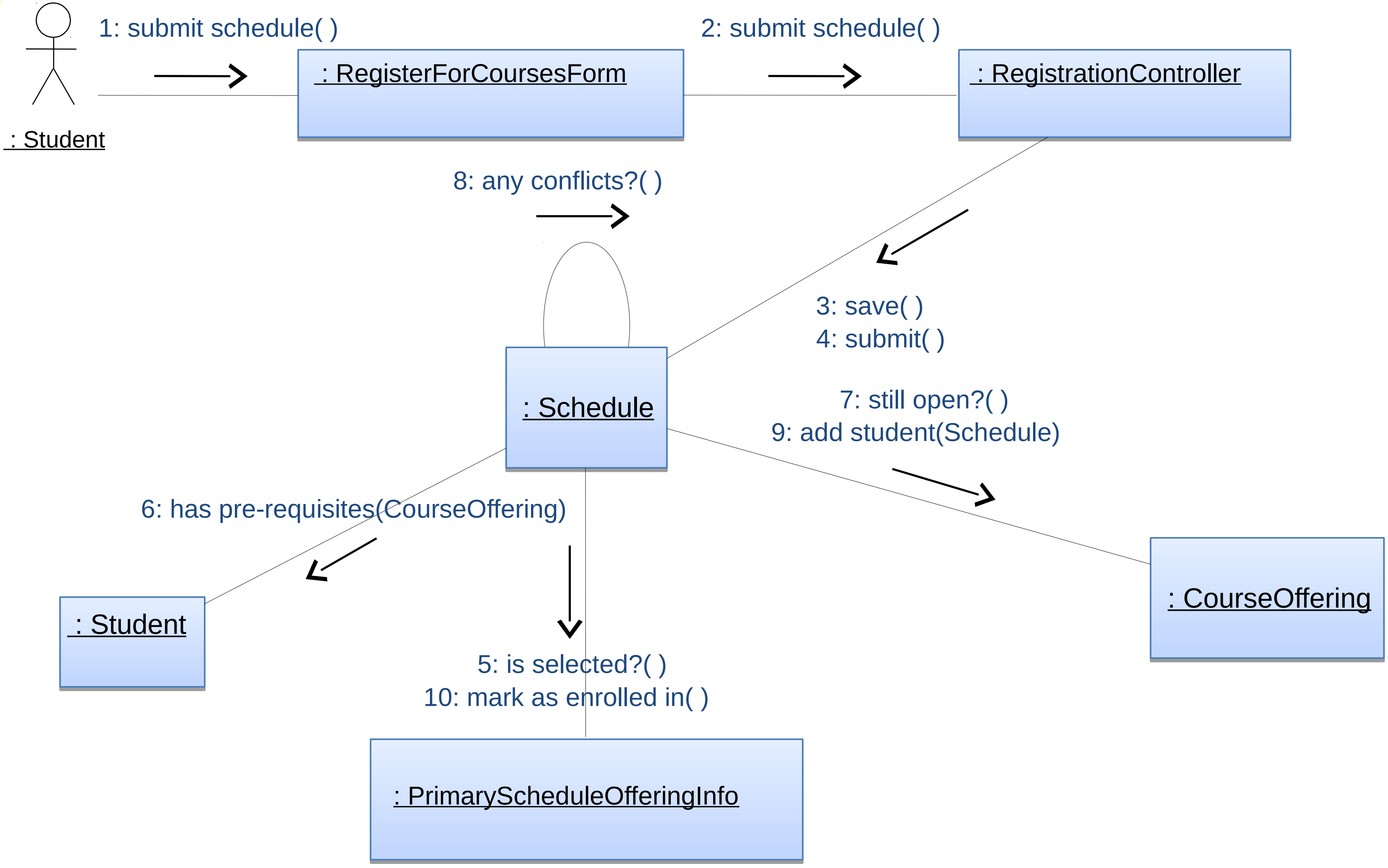
Associations

- The semantic relationship between two or more classifiers that specifies connections among their instances.
- A structural relationship specifying that objects of one thing are connected to objects of another thing.



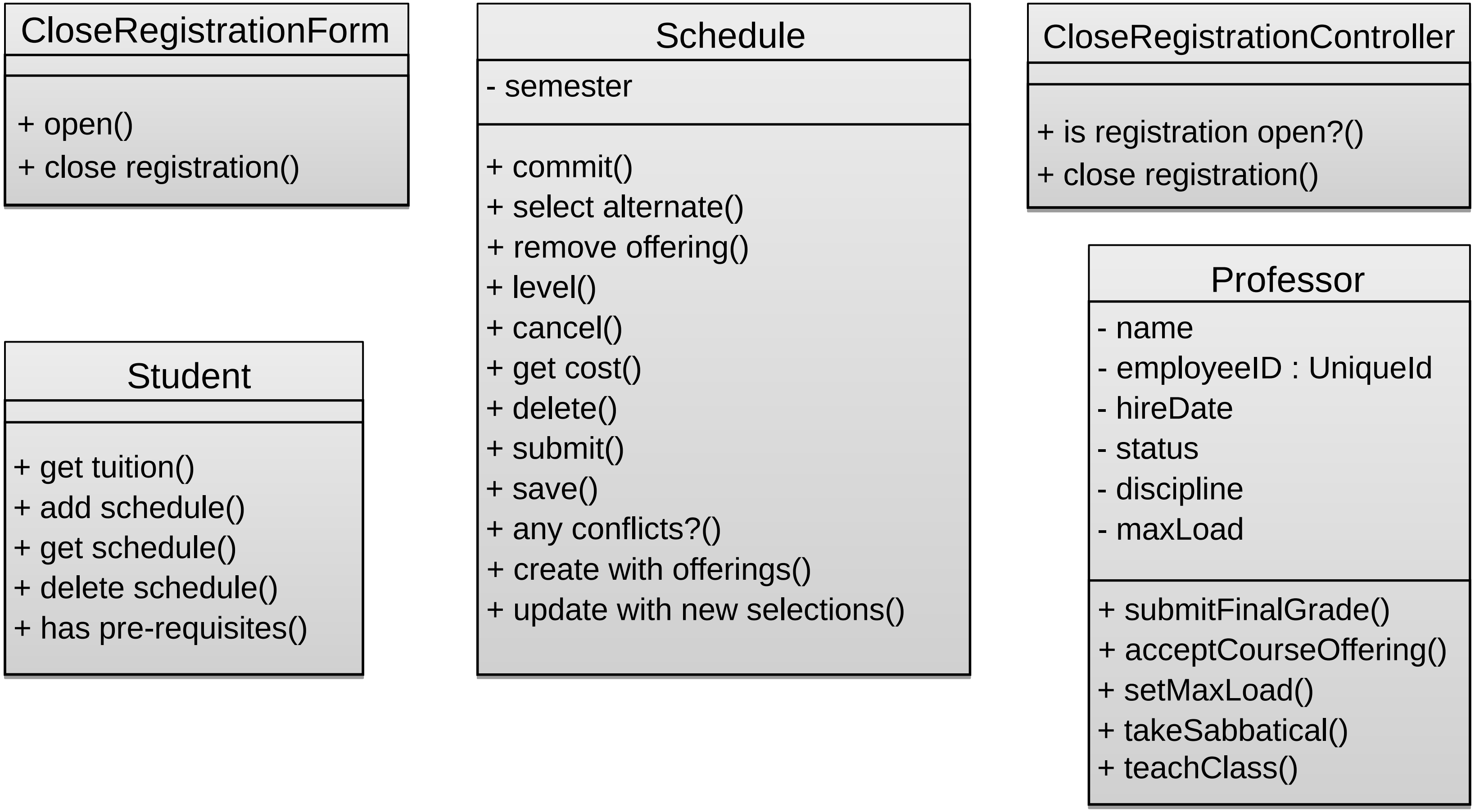
How to decide which associations? Which methods?

Use interactions to find associations



Structural Diagrams

- Static view of a system



CloseRegistrationForm

+ open()
+ close registration()

Schedule

- semester

+ commit()
+ select alternate()
+ remove offering()
+ level()
+ cancel()
+ get cost()
+ delete()
+ submit()
+ save()
+ any conflicts?()
+ create with offerings()
+ update with new selections()

CloseRegistrationController

+ is registration open?()
+ close registration()

Student

+ get tuition()
+ add schedule()
+ get schedule()
+ delete schedule()
+ has pre-requisites()

Professor

- name
- employeeID : UniqueId
- hireDate
- status
- discipline
- maxLoad

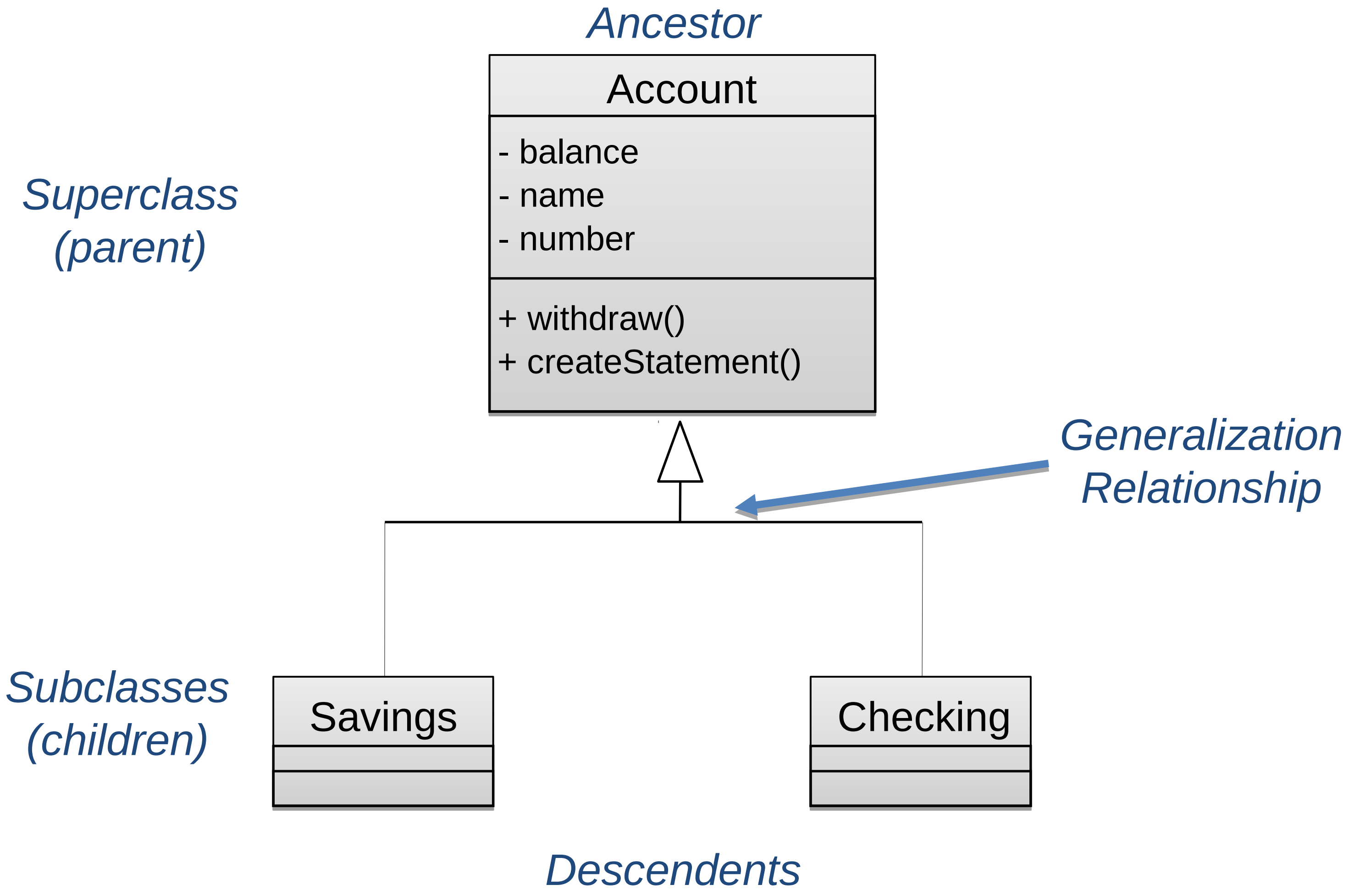
+ submitFinalGrade()
+ acceptCourseOffering()
+ setMaxLoad()
+ takeSabbatical()
+ teachClass()

Generalization

- A relationship among classes where one class shares the structure and/or behavior of one or more classes.
- Defines a hierarchy of abstractions where a subclass inherits from one or more super classes.
  - Single inheritance
  - Multiple inheritance
- Is an “is a kind of” relationship.

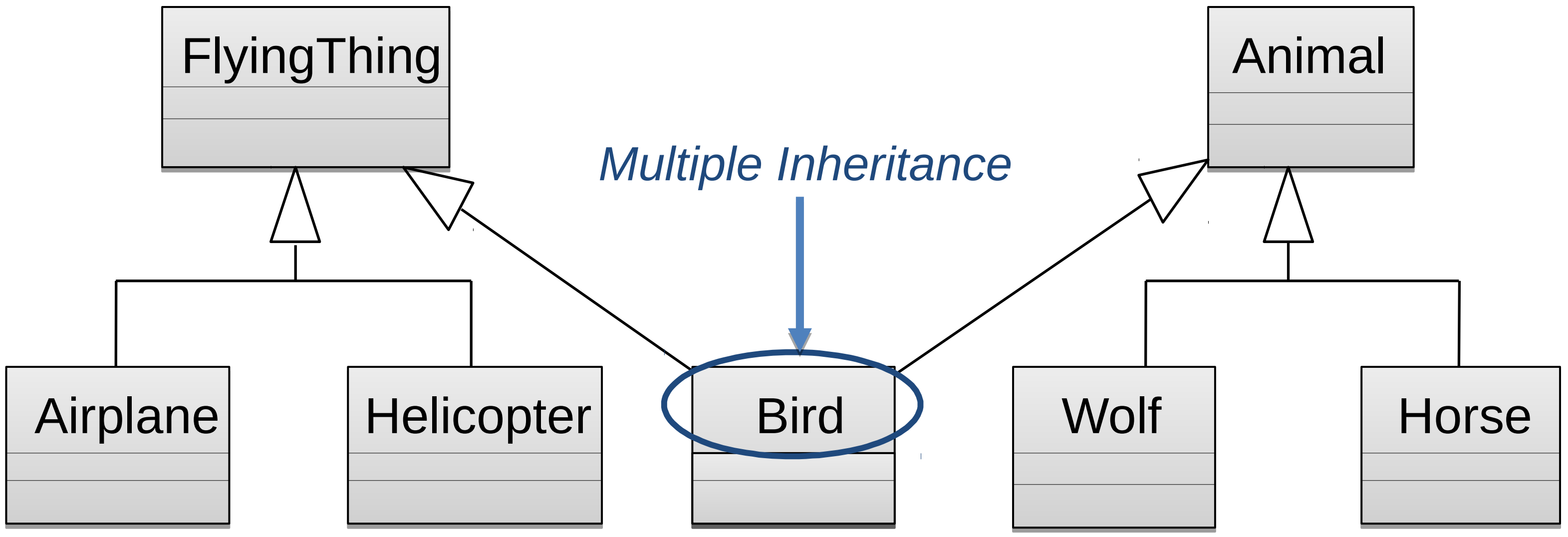
Example: Single Inheritance

- One class inherits from another.



Example: Multiple Inheritance

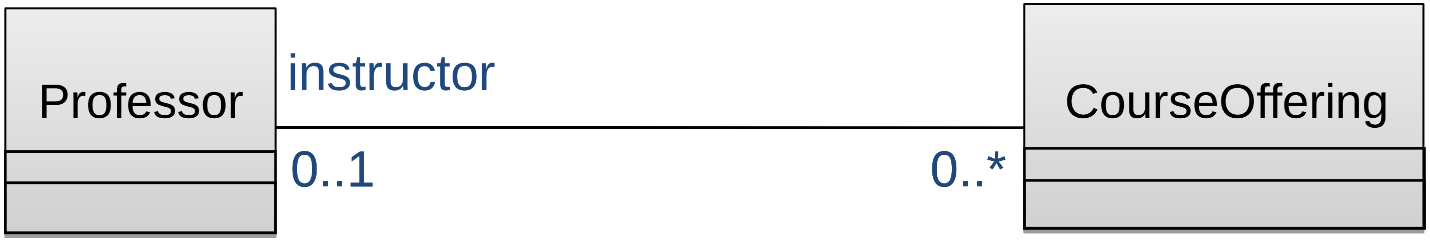
- A class can inherit from several other classes.



Use multiple inheritance only when needed and always with caution!

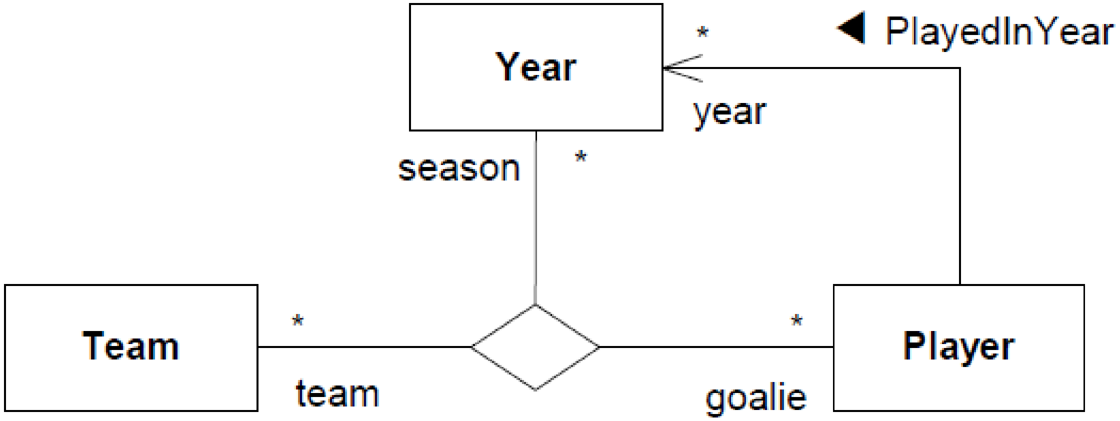
Multiplicity

- Multiplicity is the number of instances one class relates to ONE instance of another class.
- For each association, there are two multiplicity decisions to make, one for each end of the association.
  - For each instance of Professor, many Course Offerings may be taught.
  - For each instance of Course Offering, there may be either one or zero Professor as the instructor.

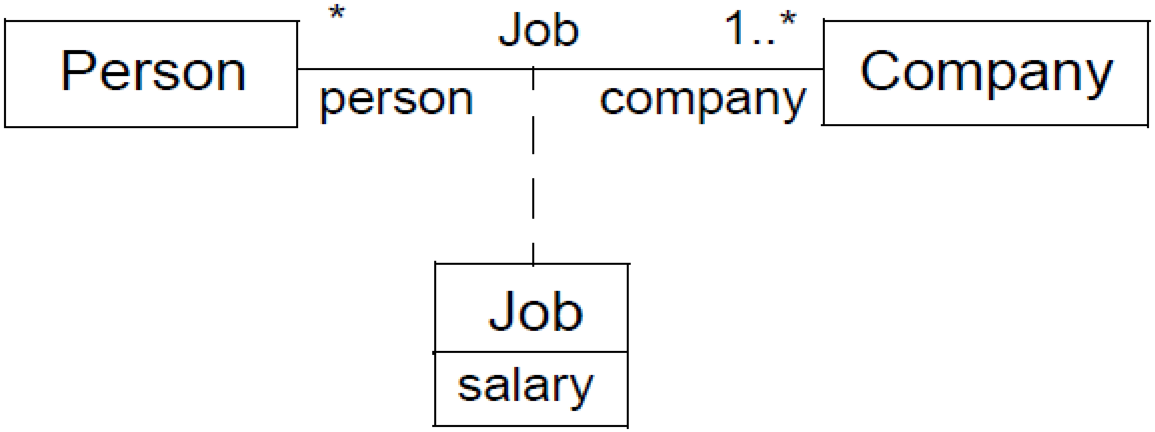


Complex Associations

- Ternary association

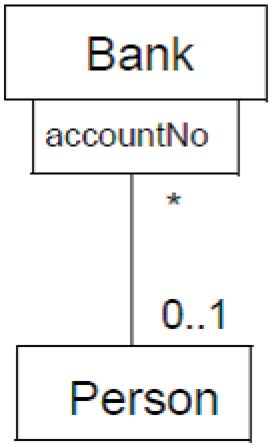


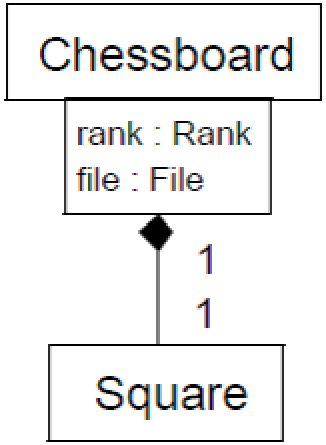
- Association Classes



Complex Associations

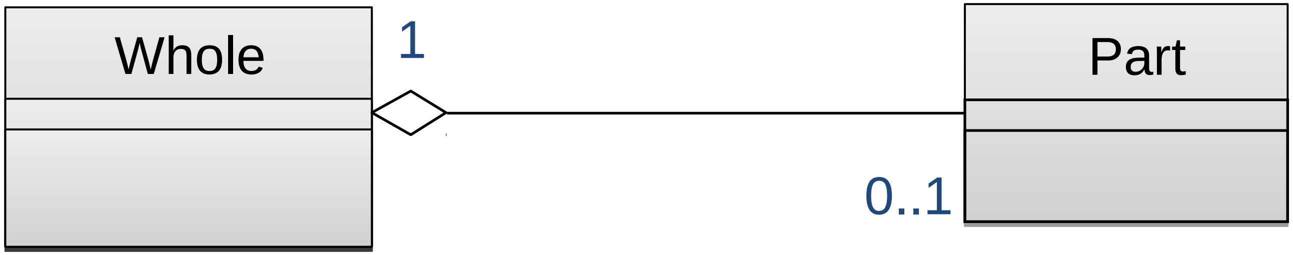
- Qualified Associations





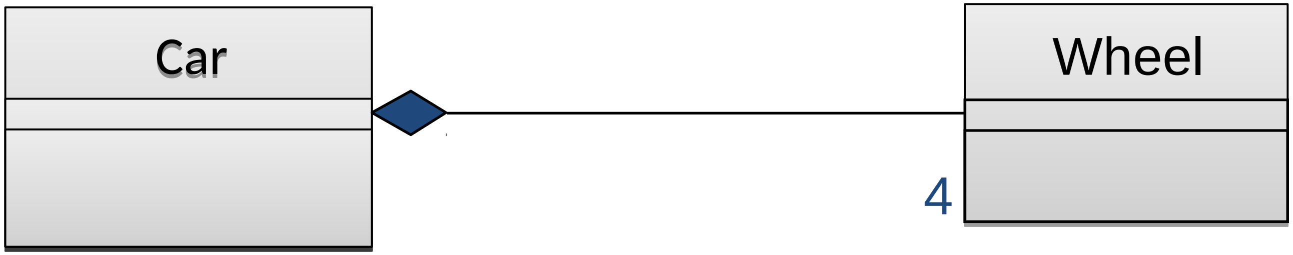
Aggregation

- A special form of association that models a whole-part relationship between the aggregate (the whole) and its parts.
  - An aggregation is an "is a part-of" relationship.
- Multiplicity is represented like other associations.



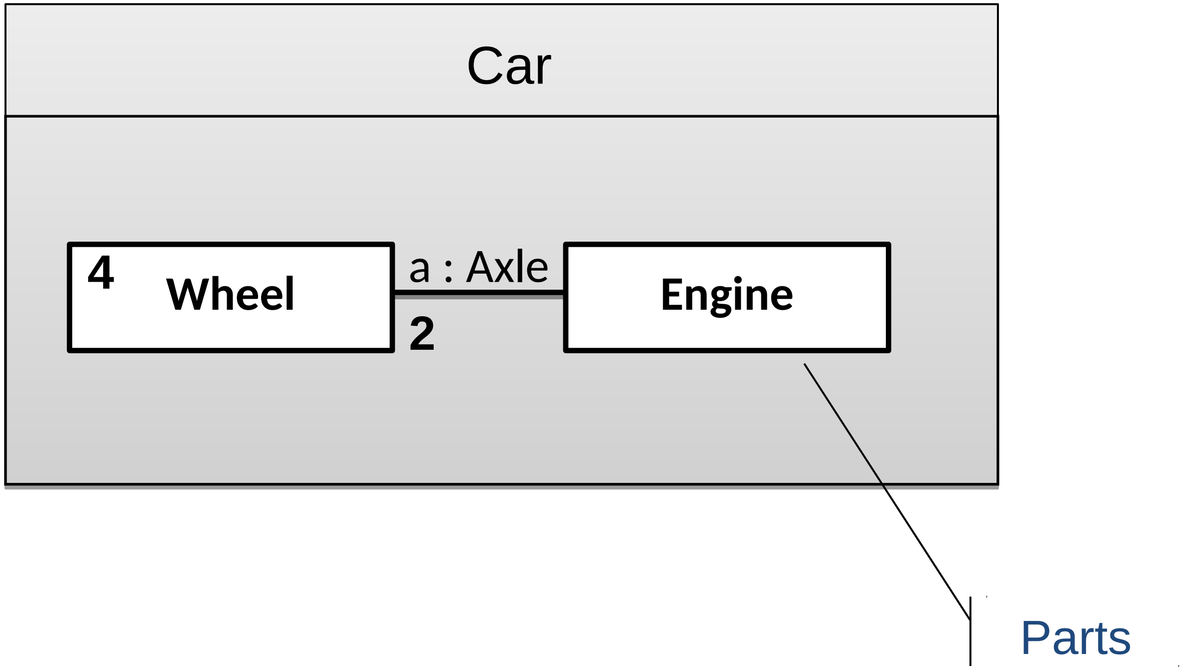
Composition

- A special form of aggregation where the composite (the whole) is responsible for the existence of the parts
  - An aggregation is an “owns” relationship
- Composition is more restricting than aggregation
  - Parts (instances) are not shared
  - E.g., A wheel is not shared between two cars



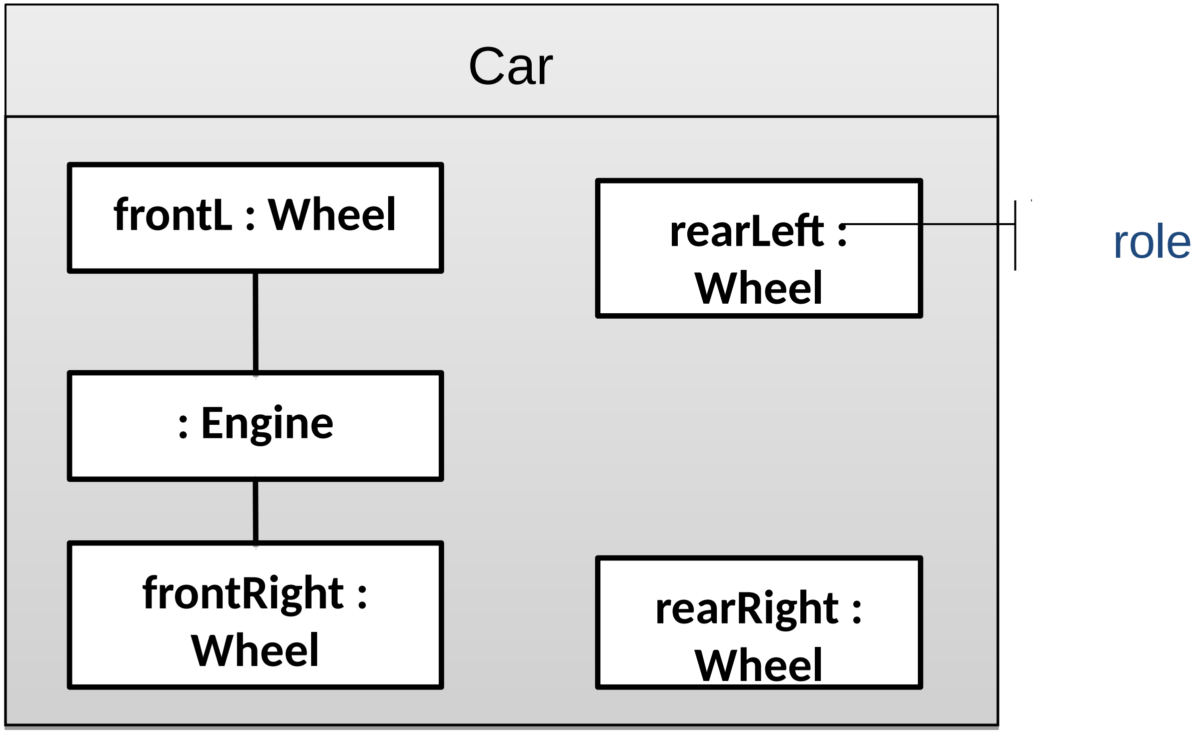
Composite Structure

- Richer and more precise way to express composition
  - Can also express aggregation (dashed lines)
  - Warning: Parts are Properties, not instances



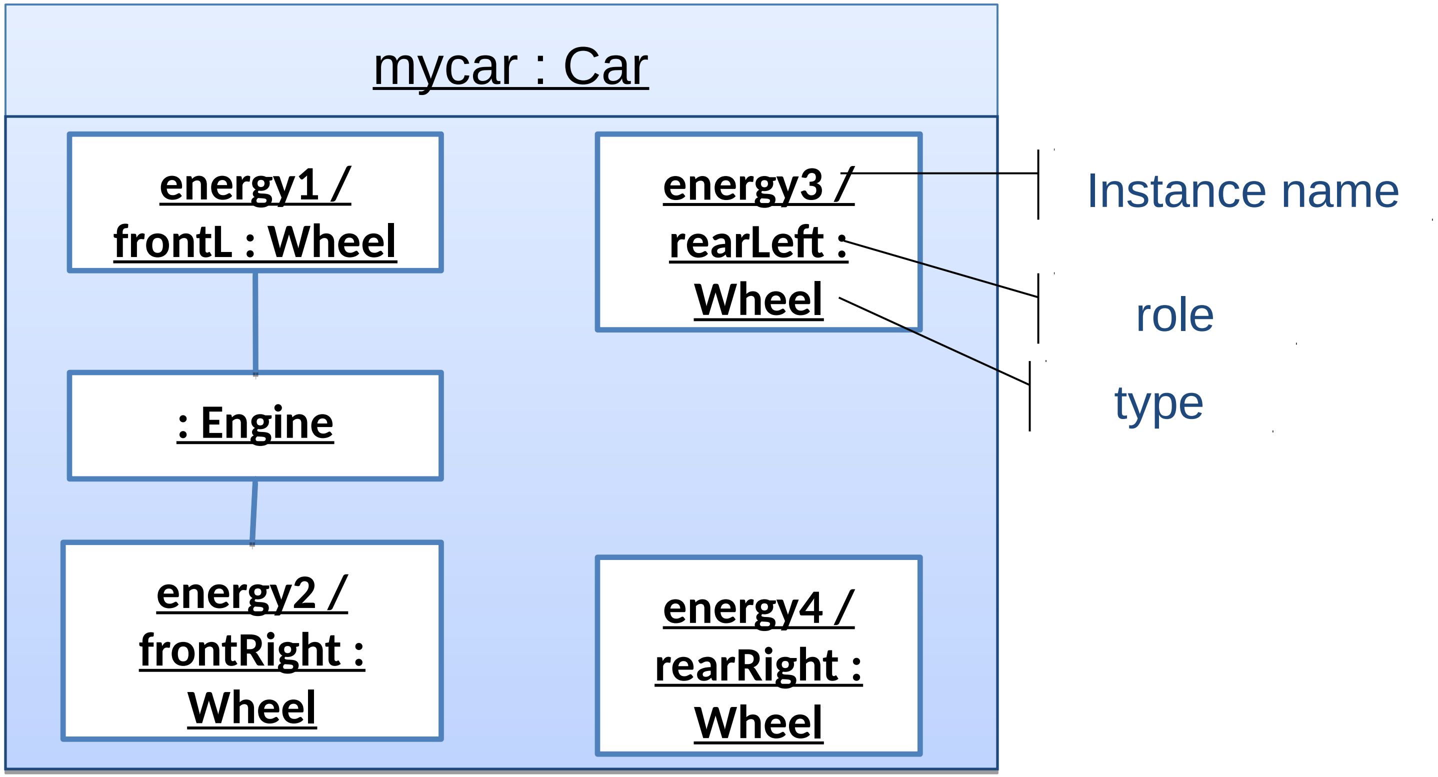
Structured classifiers

- Show roles of parts
  - Parts ARE NOT instances



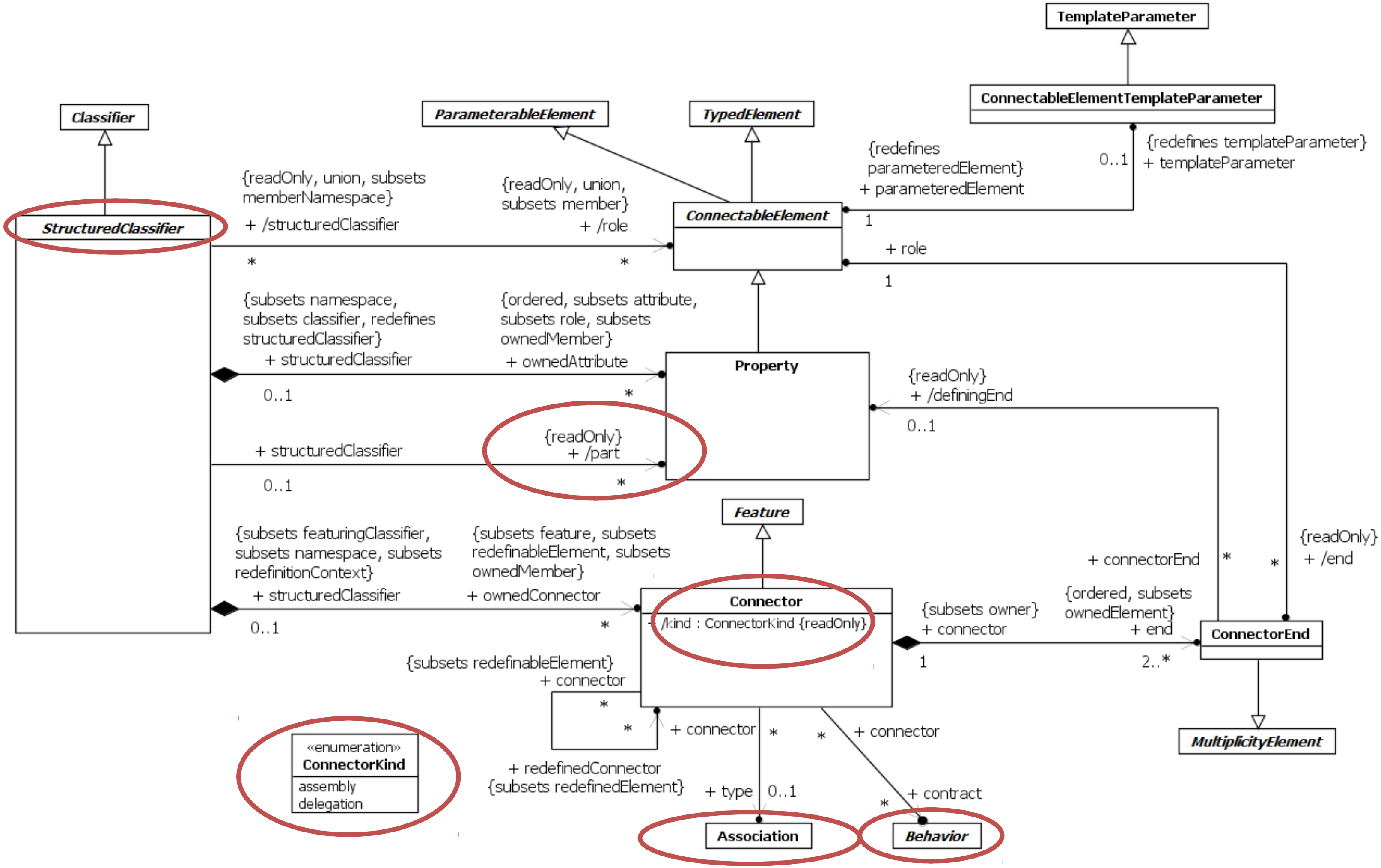
Instance of a structured classifiers

- Show roles of parts
  - Instance parts have
    - An instance name, a role and a type (optional)


mycar : Car
energy1 / frontL : Wheel
energy3 / rearLeft : Wheel
: Engine
energy2 / frontRight : Wheel
energy4 / rearRight : Wheel
Instance name
role
type

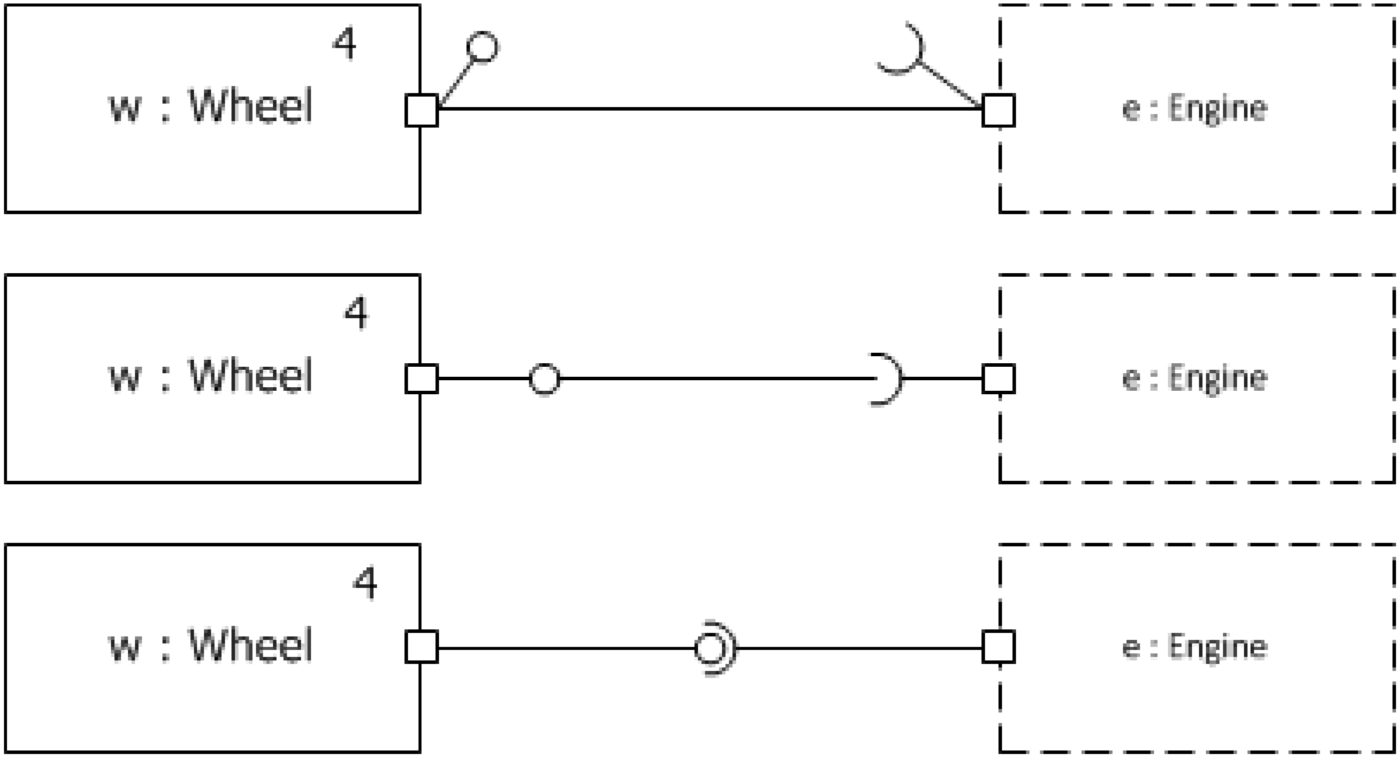
Structured Classifiers

- Metamodel



Connectors and Ports

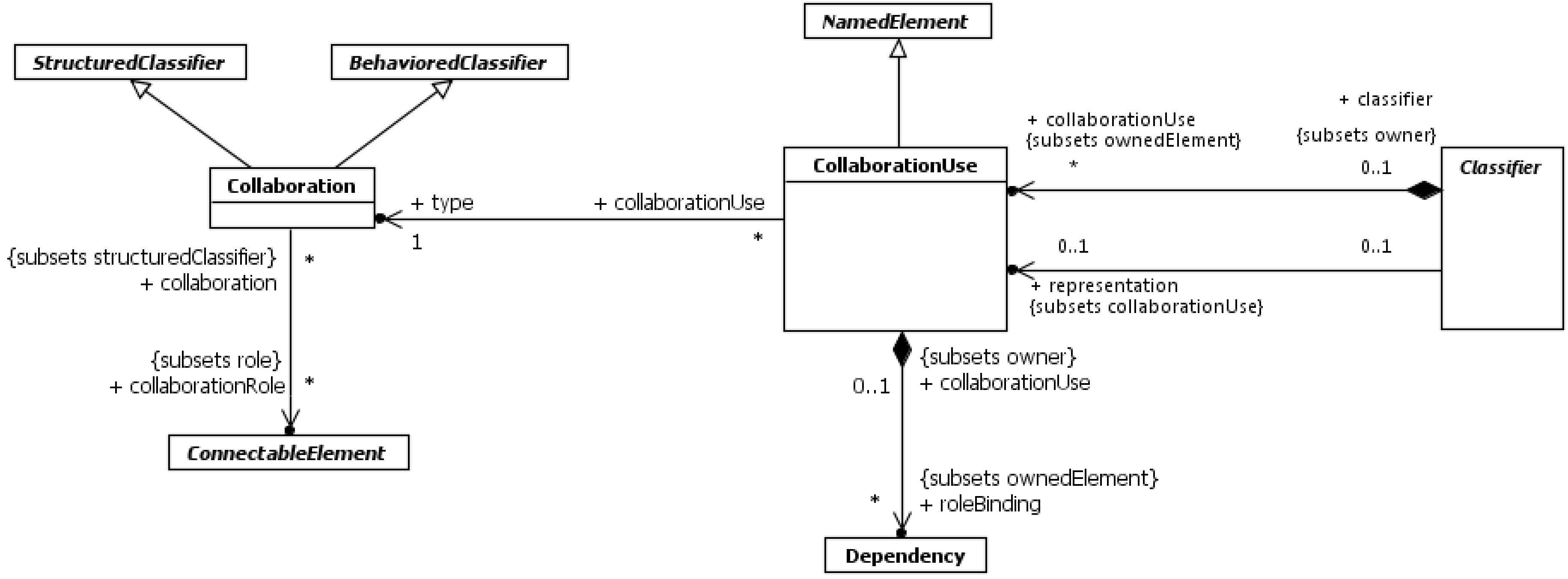
- Provided & Required Interfaces



COLLABORATIONS

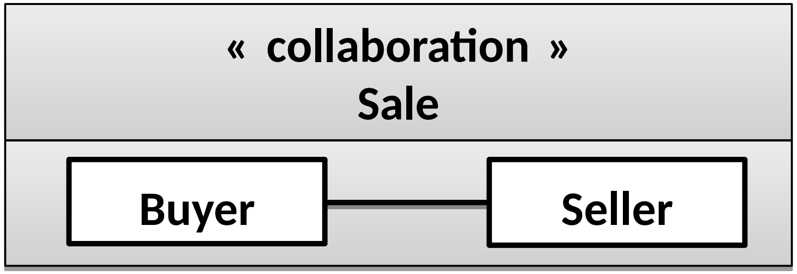
Collaborations

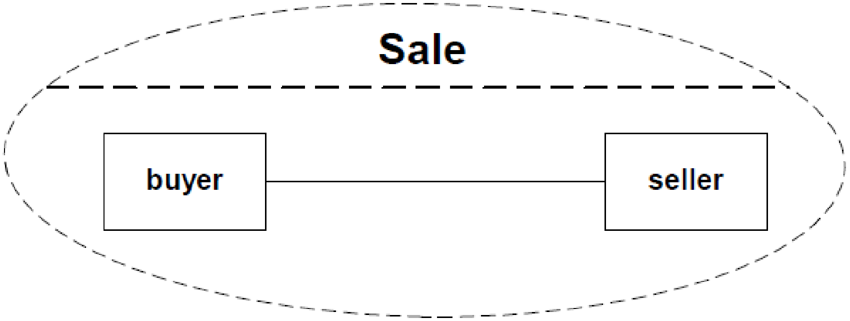
- Semantics
  - How a system of communicating elements collectively accomplish a specific task or set of tasks (pattern)
- Metamodel



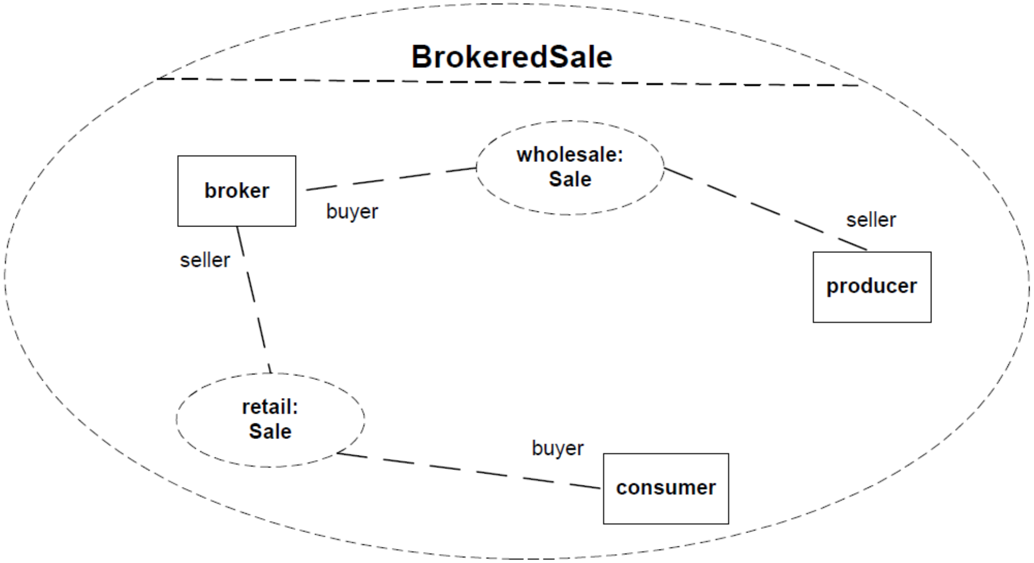
Example

- Collaboration





- CollaborationUse


BrokeredSale
broker
buyer
wholesale:
Sale
seller
producer
seller
retail:
Sale
buyer
consumer