UML language - 2

Lydie du Bousquet

Lydie.du-bousquet@imag.fr

En collaboration avec J.-M. Favre, I. Parissis, Ph. Lalanda

13 diagrams in UML 2.0

Structural diagrams

- **Class diagram**
- Object diagram
- Component diagram
- Composite structure diagram
- Deployment diagram
- Package diagram
- Profile diagram

Behavioral diagrams

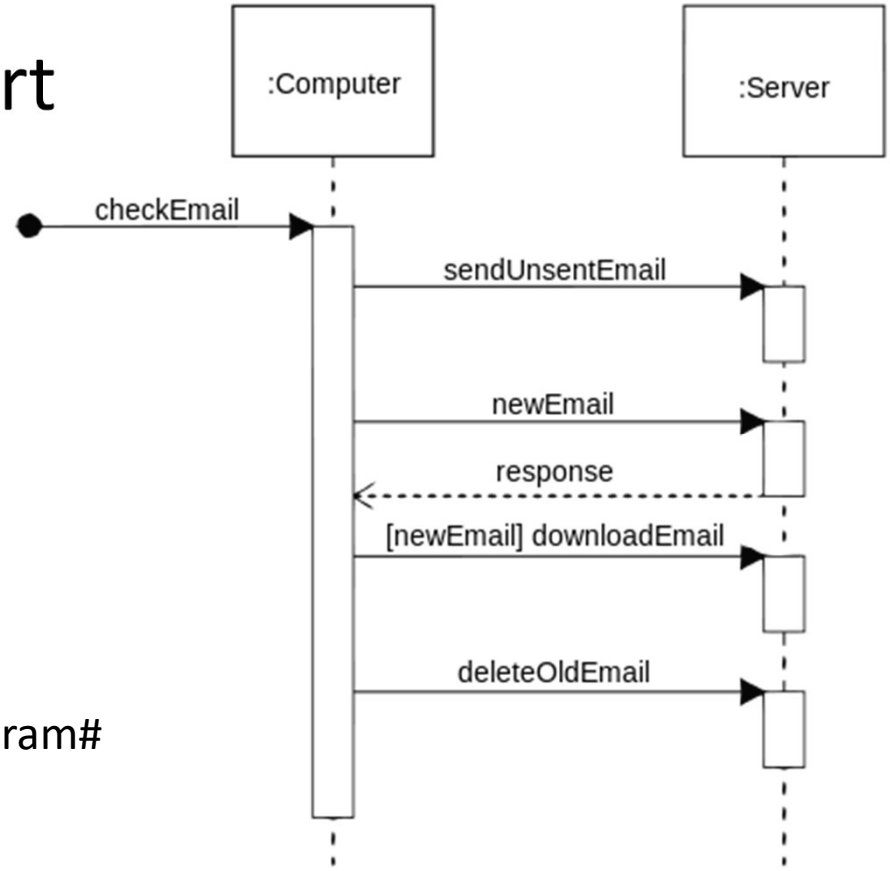
- Use case diagram
- State diagram
- Activity diagram

Interaction diagrams

- **Sequence diagram**
- Communication diagram
- Interaction overview diagram
- Timing diagram

UML sequence diagram

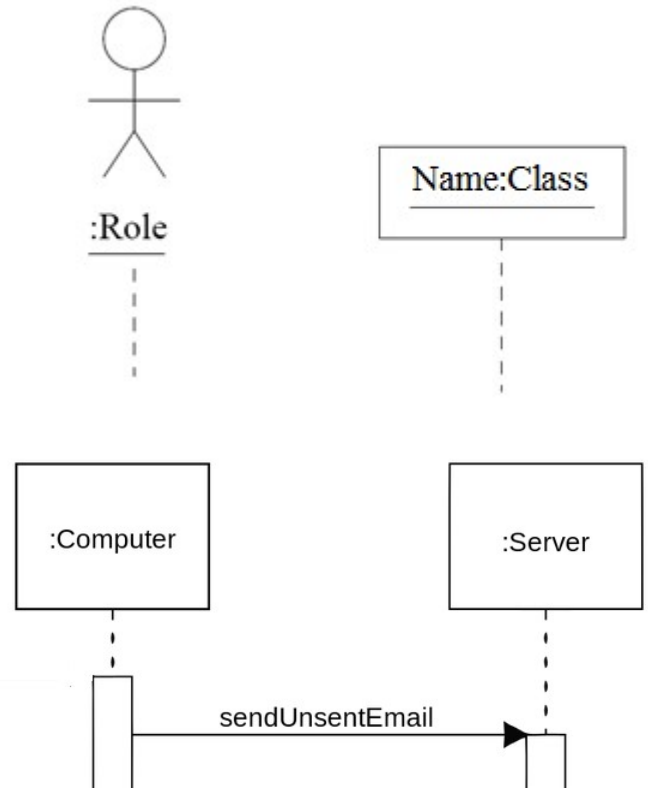
- Shows how processes operate with one another and in what order
- Message Sequence Chart



https://en.wikipedia.org/wiki/Sequence_diagram#/media/File:CheckEmail.svg

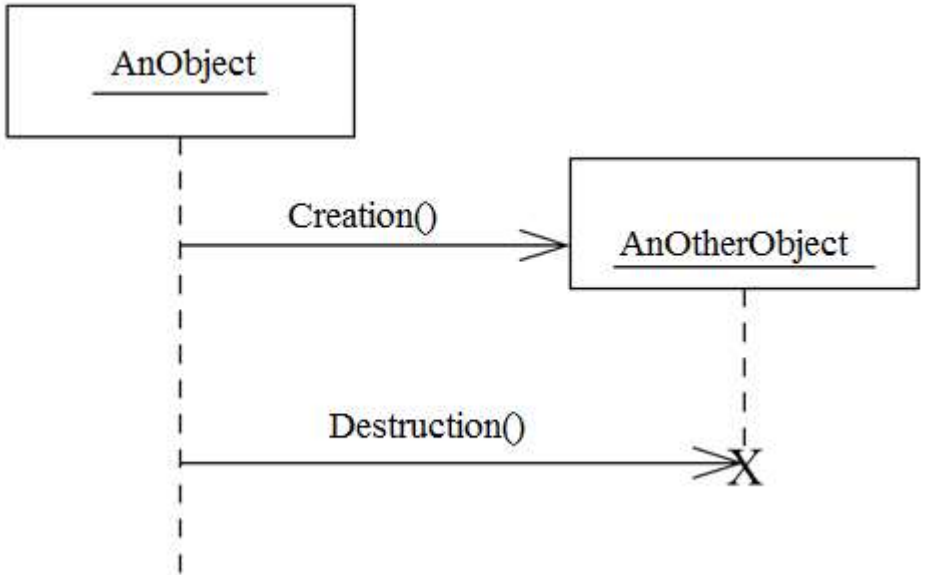
Vocabulary

- **Lifeline**
  vertical line
  represents the time
- **Message**
  horizontal arrow
  with a message name

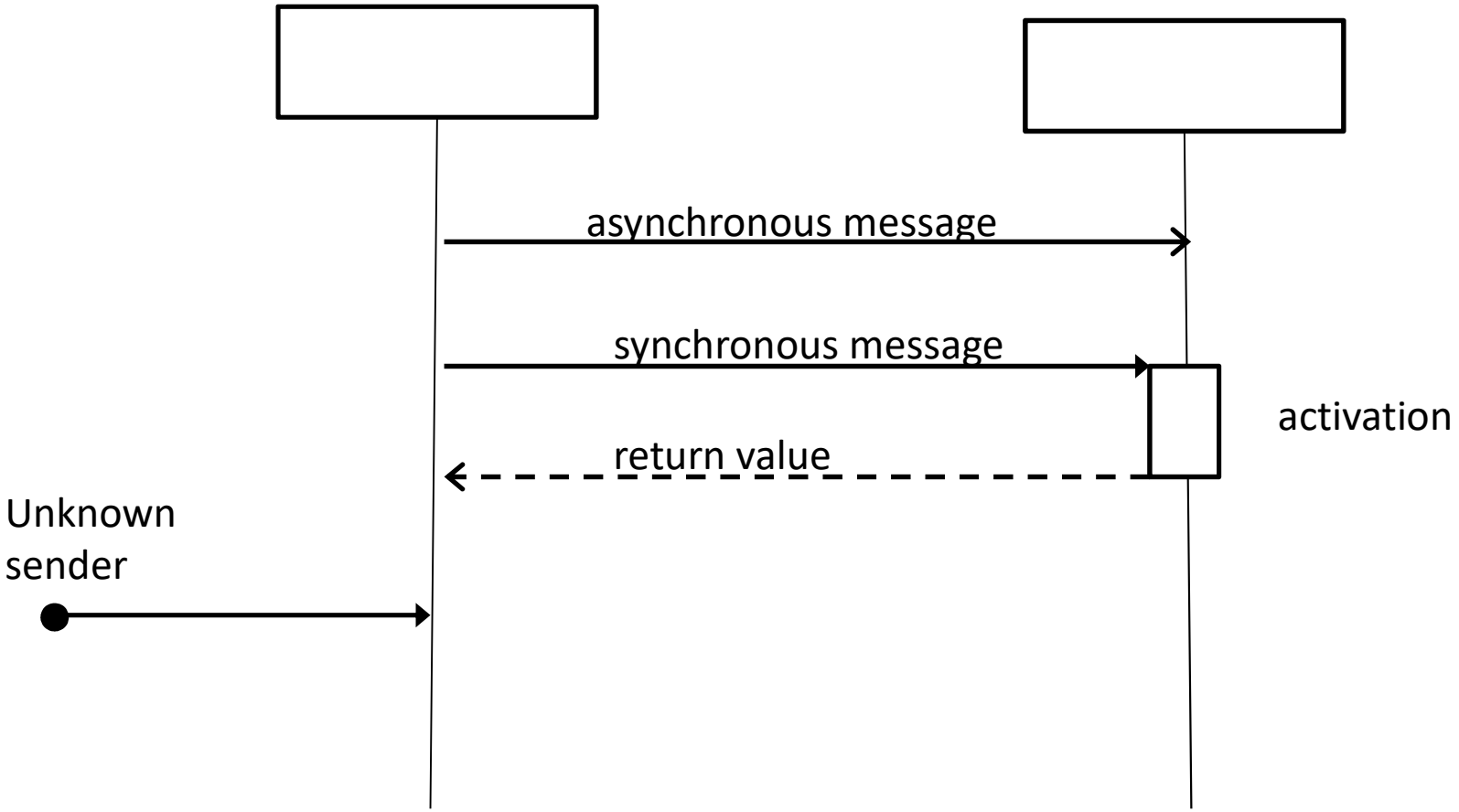


Messages for object creation and destruction

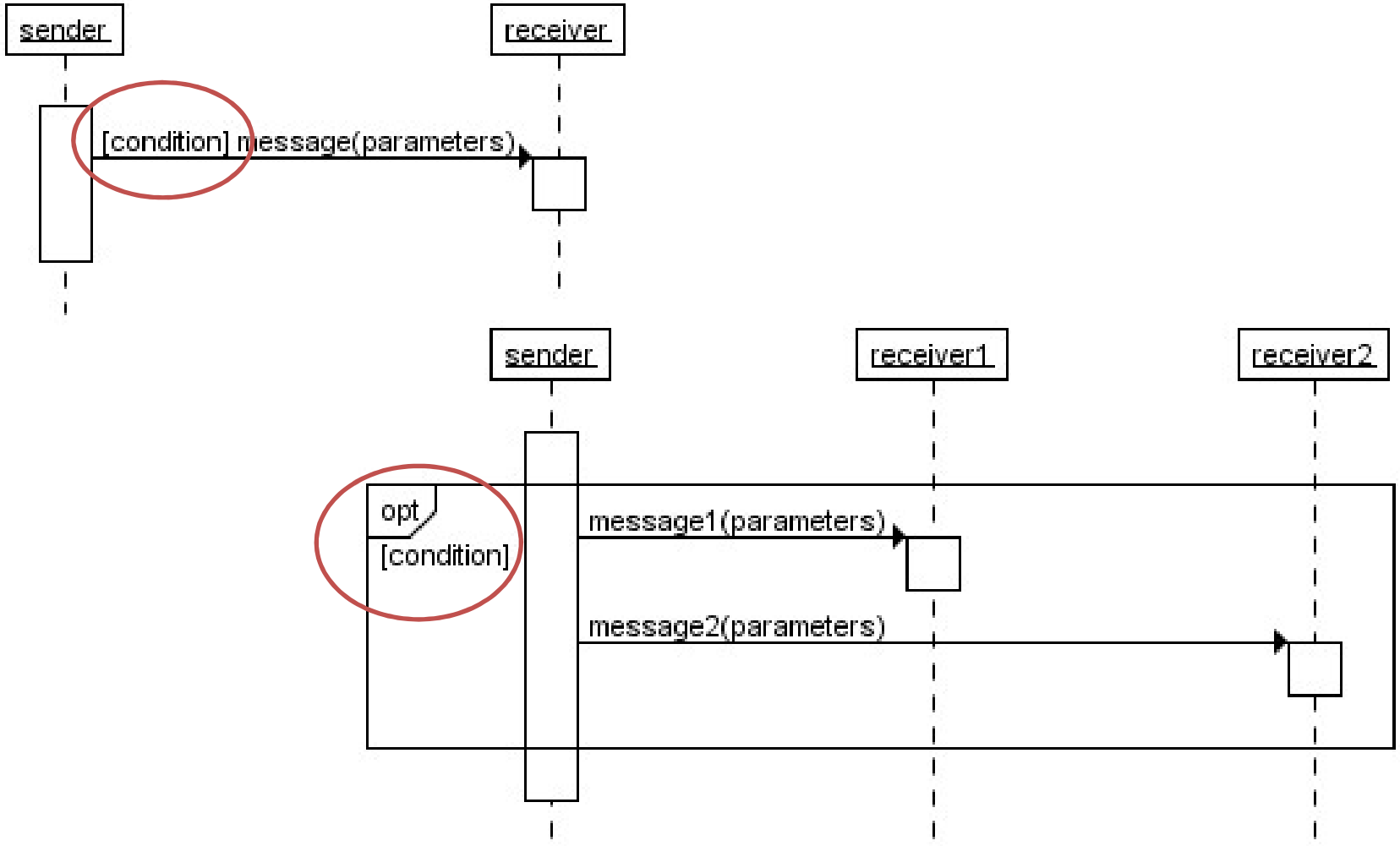
- **Creation**()
  causes the creation of an object
- **Destruction**()
  causes the destruction of an object,
  a large X is used at the end of the lifetime



Type of messages

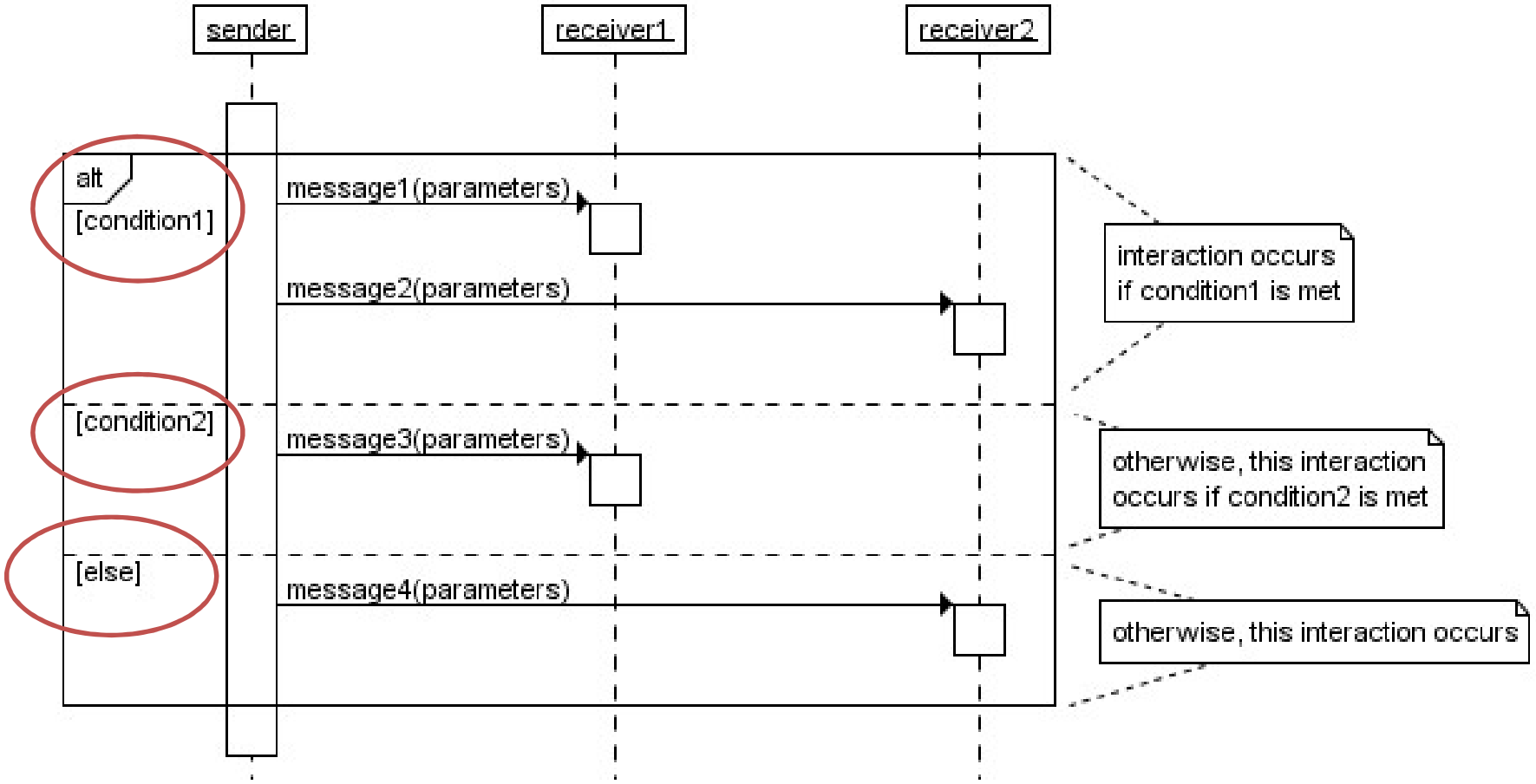


Conditional behaviors



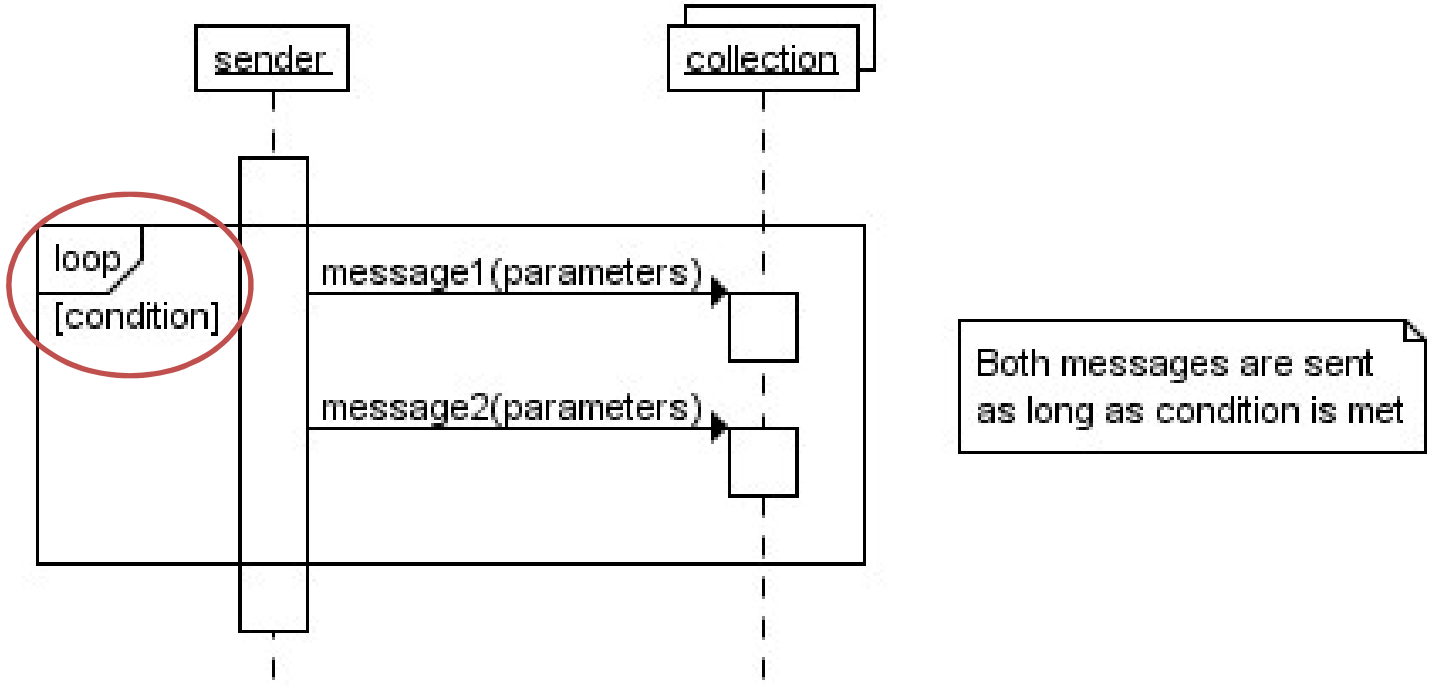
http://www.tracemodeler.com/articles/a_quick_introduction_to_uml_sequence_diagrams/

Conditional behaviors

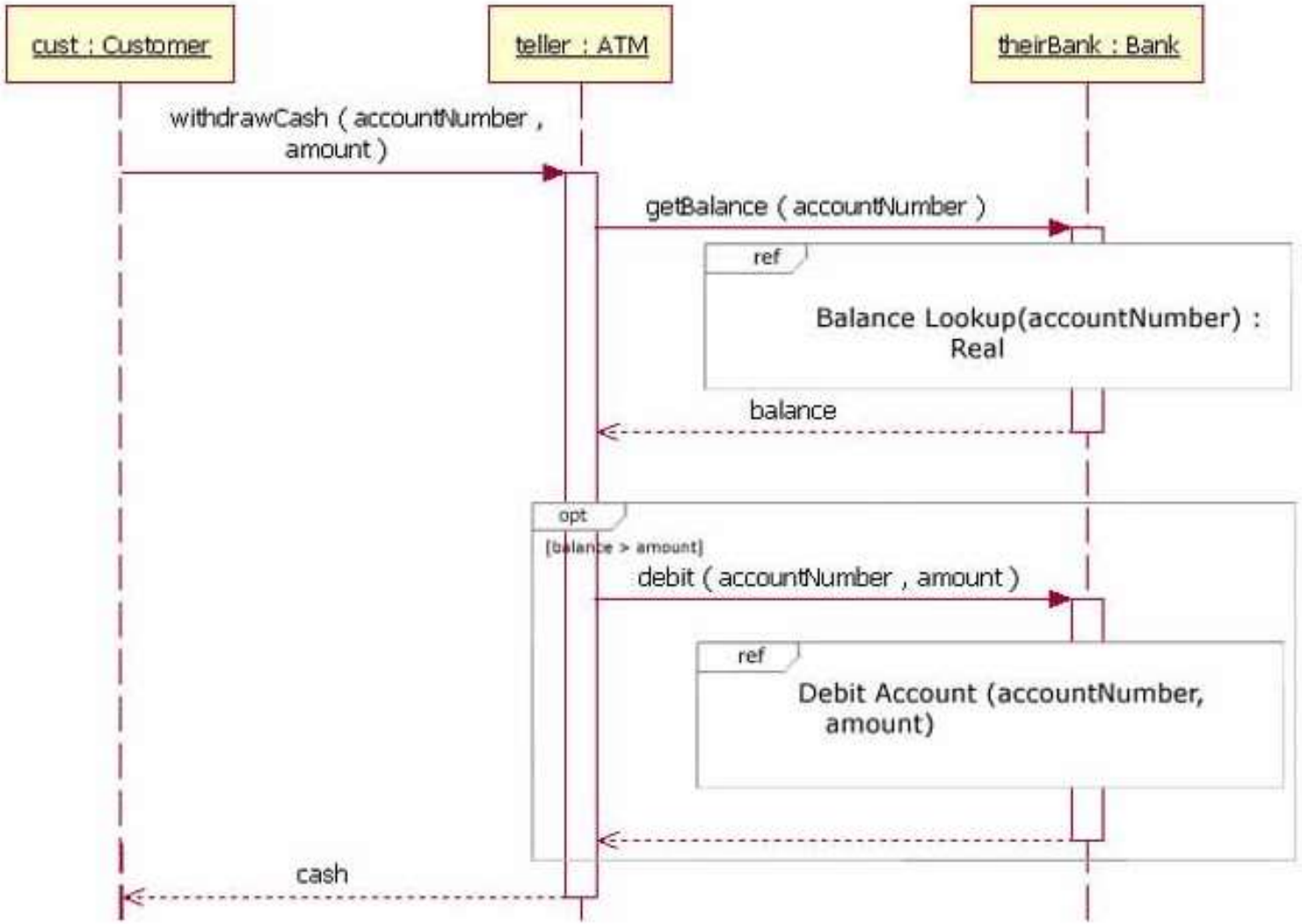


http://www.tracemodeler.com/articles/a_quick_introduction_to_uml_sequence_diagrams/

Repetition


sender
collection
loop
[condition]
message1(parameters)
message2(parameters)
Both messages are sent
as long as condition is met

Reference to other sequence diagrams



http://www.ibm.com/developerworks/rational/library/3101.html

Recommendations

- Adapt the description to your needs
  - Analysis / design…
  - Level of abstraction
- Keep simple
  - Separation of concerns
  - Choose precise scenarios
- Do as much as necessary, but no more
  - Time is precious

Exercises

Exercise:
The morning routine of Joe

The morning routine of Joe starts when his alarm clock wakes up Joe. Joe rubs his eyes. He then starts the coffee machine. While the coffee machine is heating, Joe switches one the TV to watch the news. He gets his cup of coffee next.

Q. Represent this routine as a sequence diagram

13 diagrams in UML 2.0

Structural diagrams

- **Class diagram**
- **Object diagram**
- Component diagram
- Composite structure diagram
- Deployment diagram
- Package diagram
- Profile diagram

Behavioral diagrams

- Use case diagram
- State diagram
- Activity diagram

Interaction diagrams

- Sequence diagram
- Communication diagram
- Interaction overview diagram
- Timing diagram

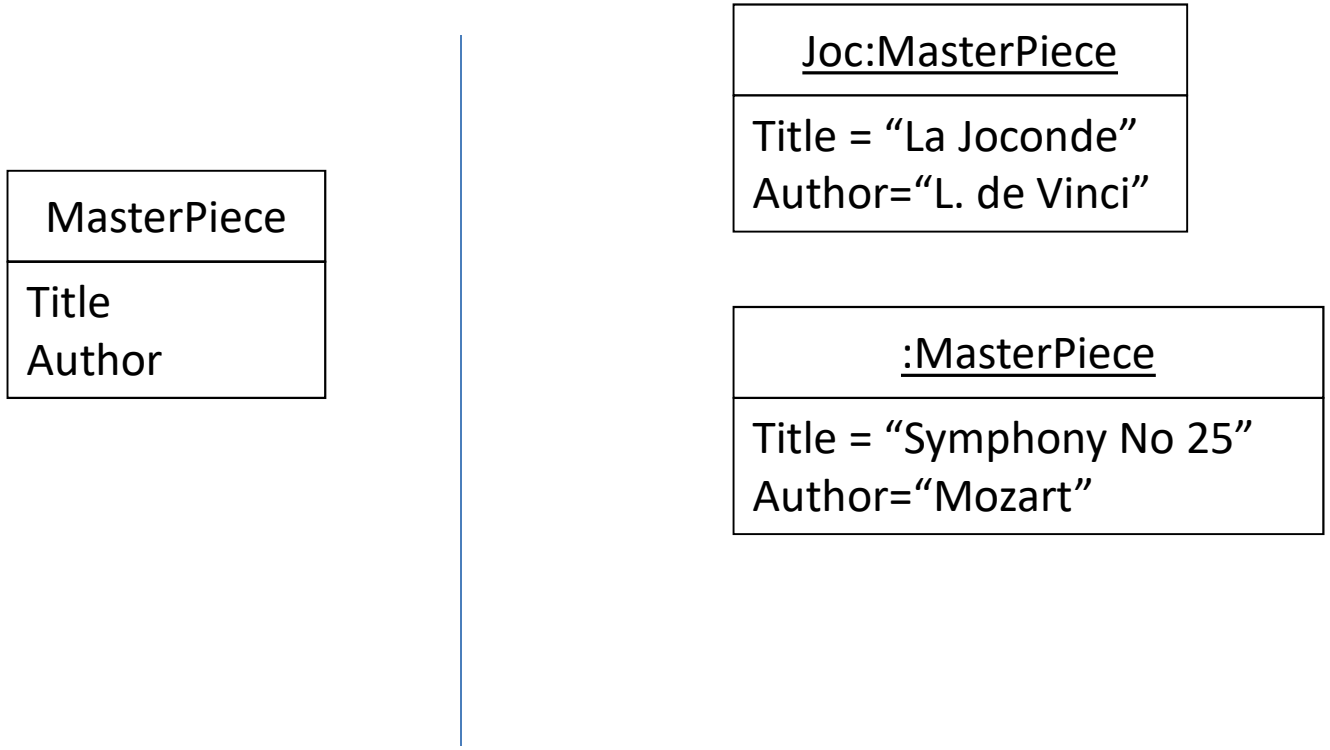
A class diagram

- Show the classes and their relations
  - Attributes and operations of the classes
  - Association, Inheritance, aggregation
- Abstract dynamical and temporal aspects
- Can be instantiated as object diagram
- Is used for
  - Analysis, Design
  - Documentation

An object diagram

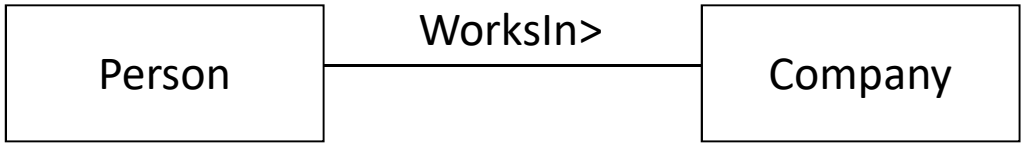
- Show **objects** and their relations : snap of running
  - Attributes and their values
- Is used for
  - Making the prototype of a system
  - Reverse engineering
  - Modeling complex data structures.
  - Understanding the system from practical perspective.
  - Validation or creation of the class diagrams

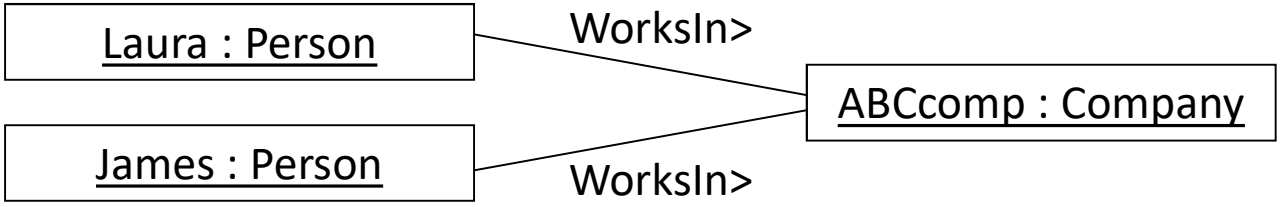
Class vs Object



Associations

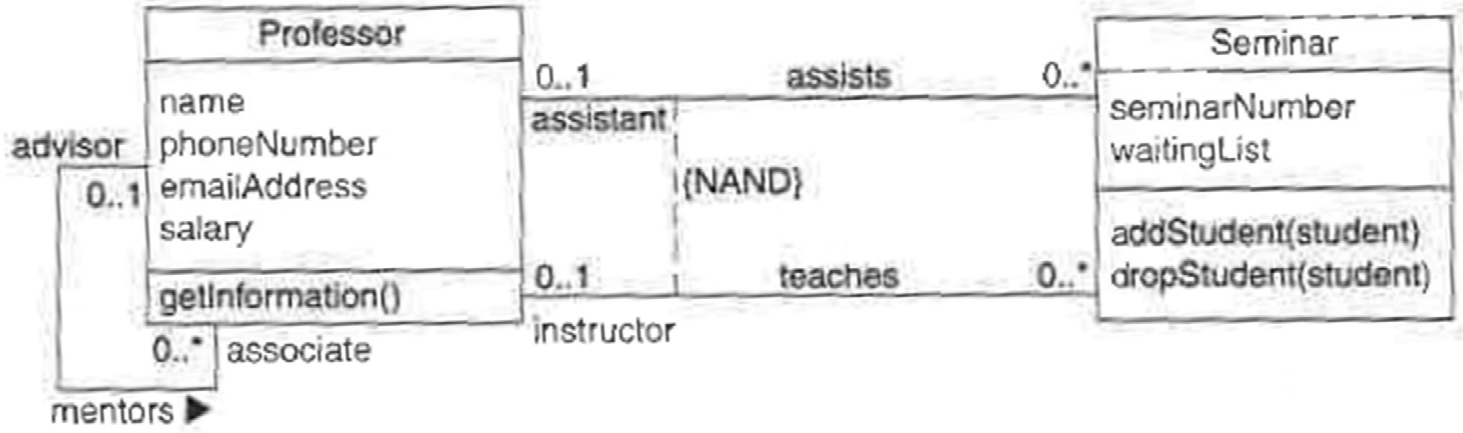
- Both for class and object diagrams
- Labeled with a name
- Arrow to indicate the direction in with text should be read





Role names

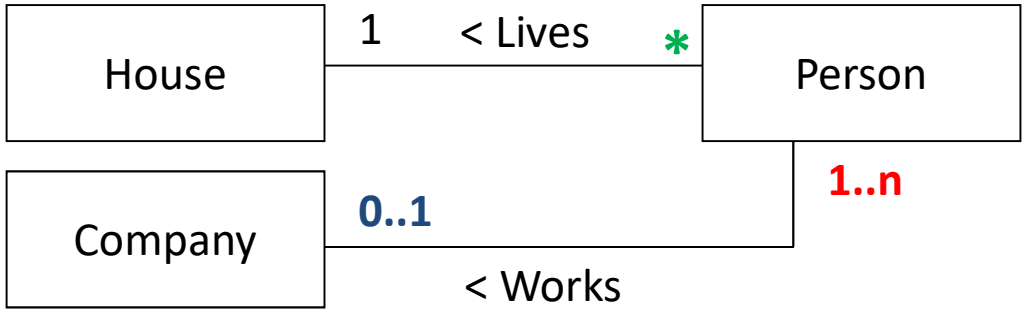
- Are Indicated on the association ends
- Precise how a class is involved in the association
- Are optional



Example from "The elements of UML 2.0 style", Scott Ambler

Multiplicity

Indicates the number of object instances at the far end of an association for an instance of a class



A person works for at **most one** company

Several persons work for a given company (and **at least one**)

The number of persons that lives in a house is **unknown**

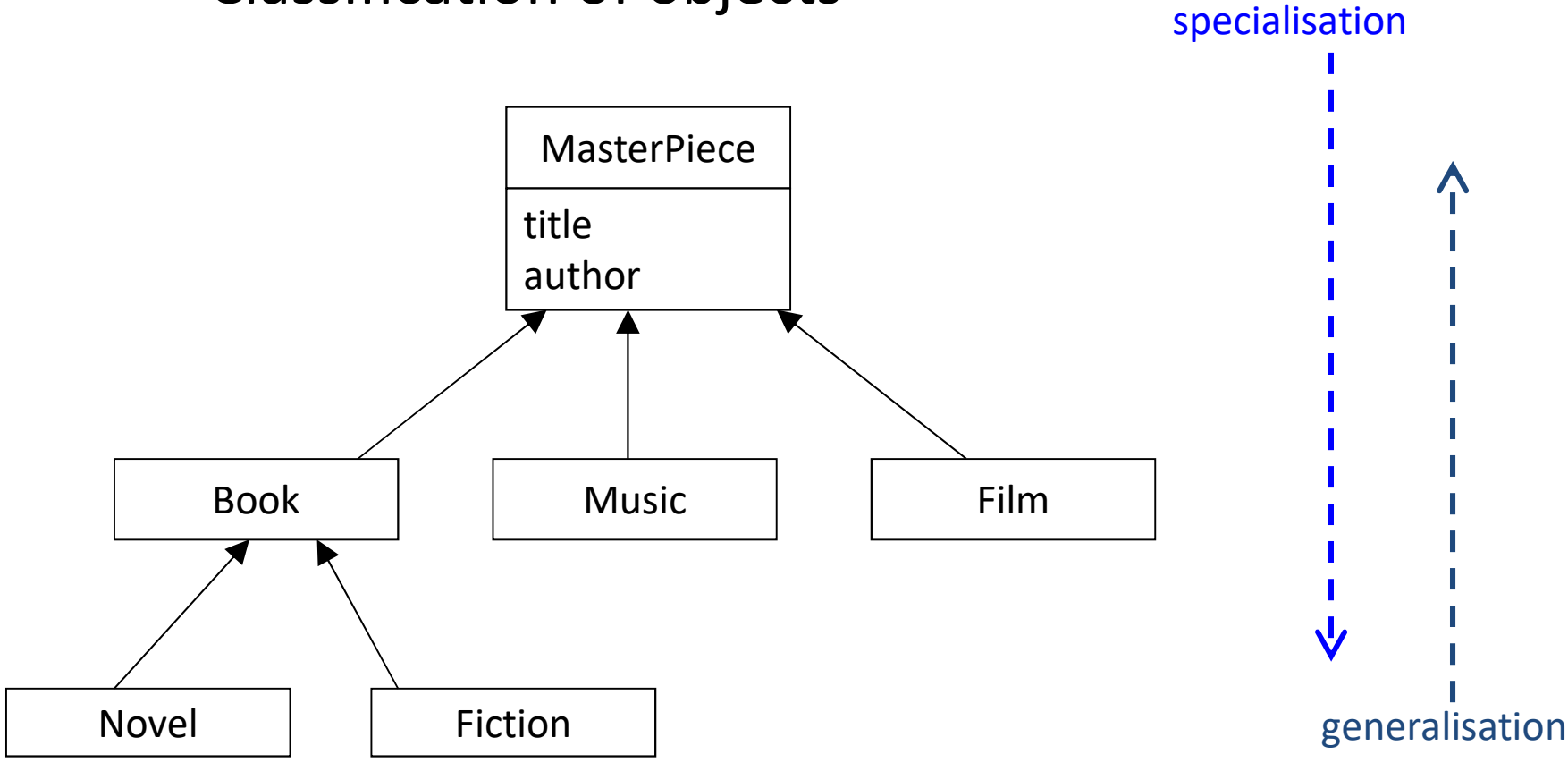
Multiplicity

- **n** : exactly « n » (integer) (3 ou 12 …)
- **n .. m** : between« n » and « m »

  0..1, 3..n, 1..31
- ***** : means « 0..n » or « 0..* »
- **n .. *** : means « n » or more

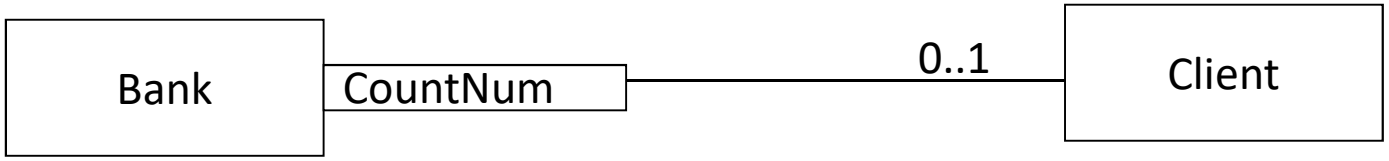
Inheritance

- To reuse and deal with complexity
- Classification of objects

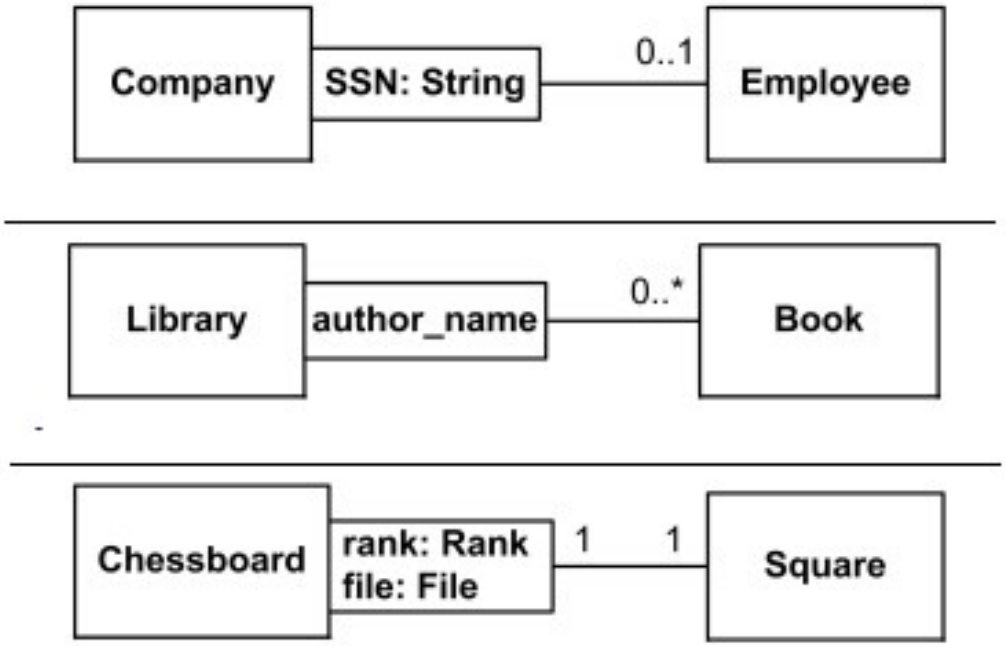


Association Qualifier

- To define a partition of the set of associated instances with respect to an instance at the qualified end.



Association Qualifier (examples)

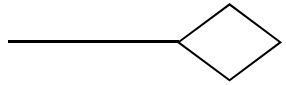


At most one

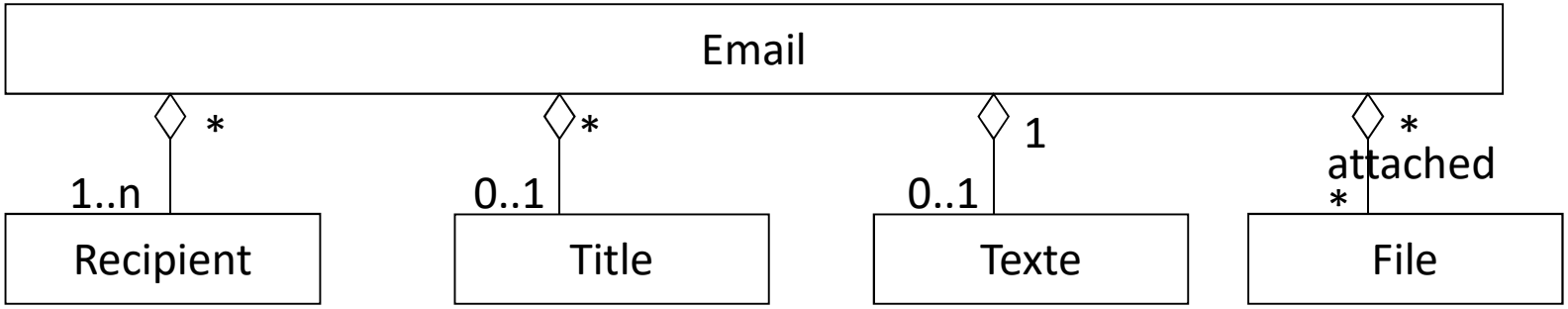
None to many

Exactly one

Aggregation

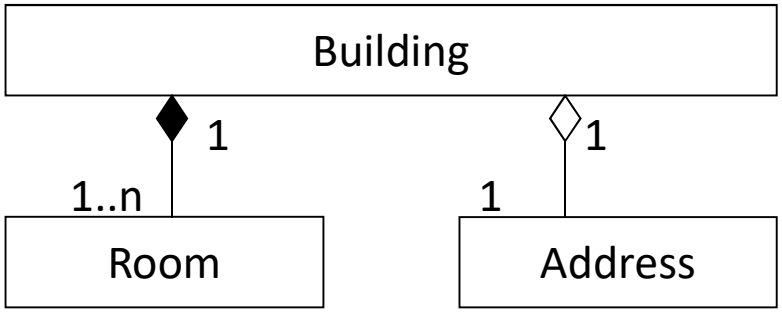


- Non symmetrical association
- Variant of the "has a" association relationship
- « Set / collection /container » of elements (other classes)
- Set and elements can exist independently =
  contents of the container are not automatically destroyed when the container is.



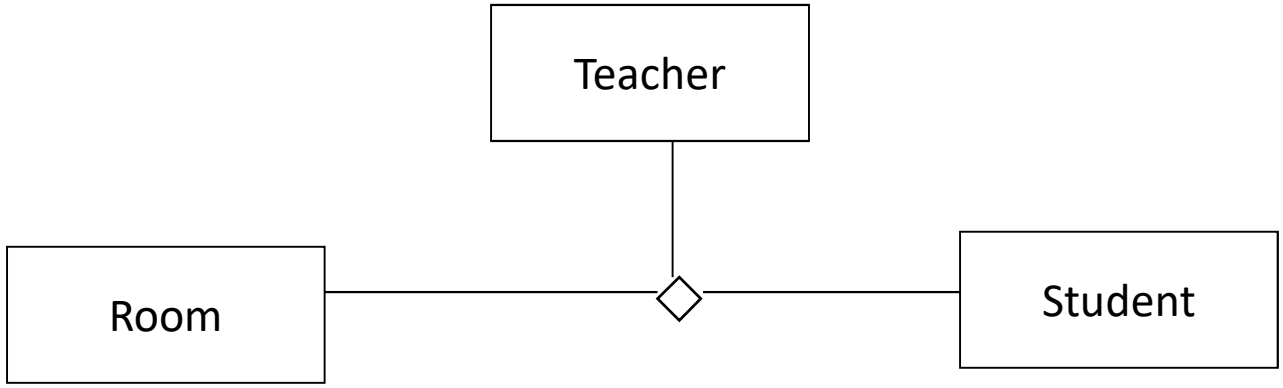
Composition

- Strong aggregation
- Whole and parts are linked:
  if the whole is destroyed, parts are destroyed as well



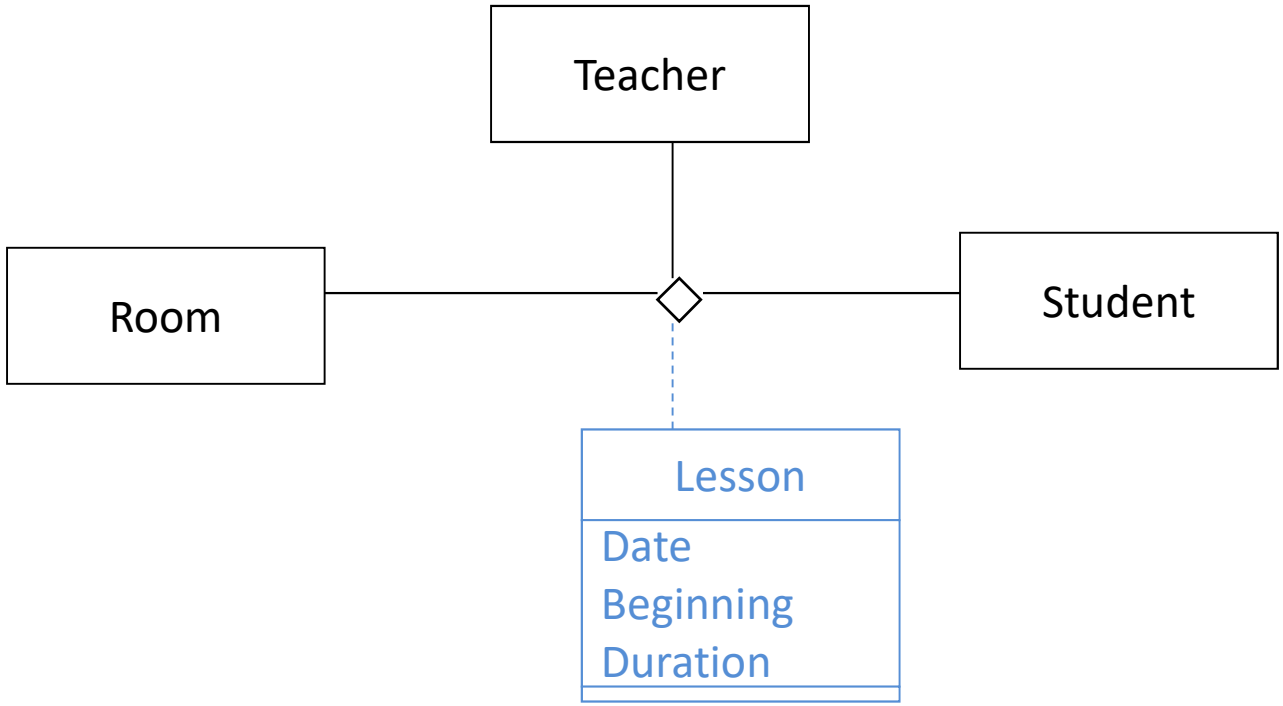
N-ary Association

- To relate three or more classes



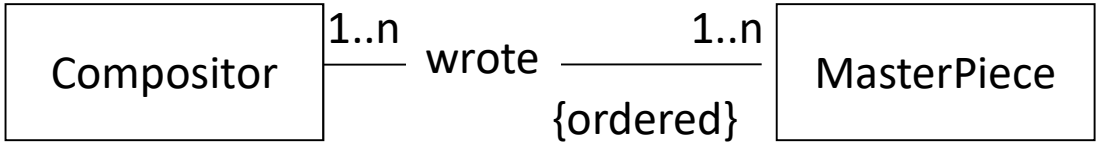
Association class

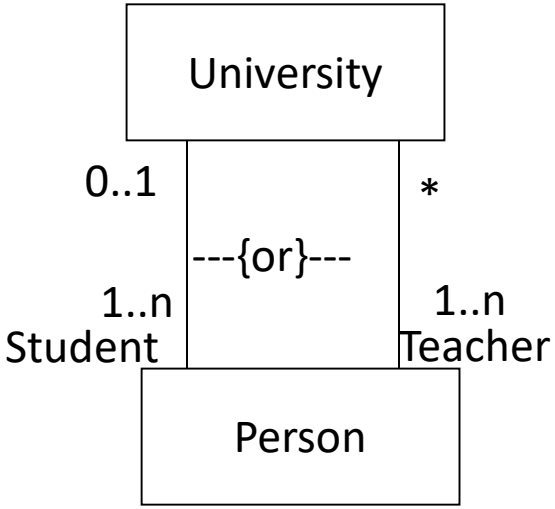
- To specify a relation



Constraint on an association

- Expression pour préciser le rôle ou la portée d'un élément de modélisation
- Peut-être exprimé en langage naturel ou en OCL



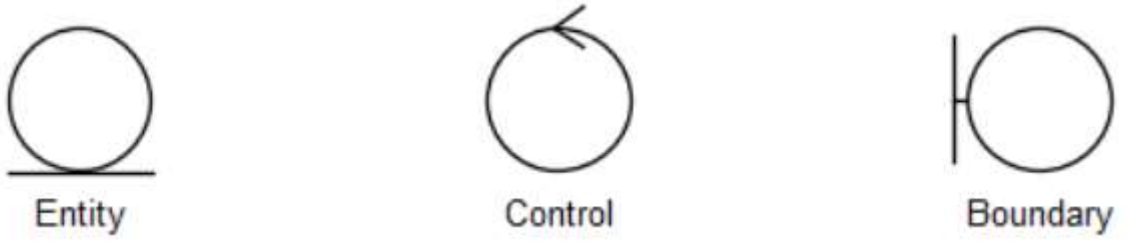


From analysis to design

- The “style” of the class diagram (= level of description) depends of the phase
- At the analysis: Domain class Model
  - Classes
  - Relations, roles, cardinalities
- Design level
  - the attributes, the types,
  - the visibility, …
- Never model the keys

To go further

- Abstract classes vs concrete classes
- Stereotypes for classes:
  - Boundary classes: to represent objects at the interface with actors
  - Entity classes: to represent system data
  - Control classes: manage the flow of interaction



Exercises