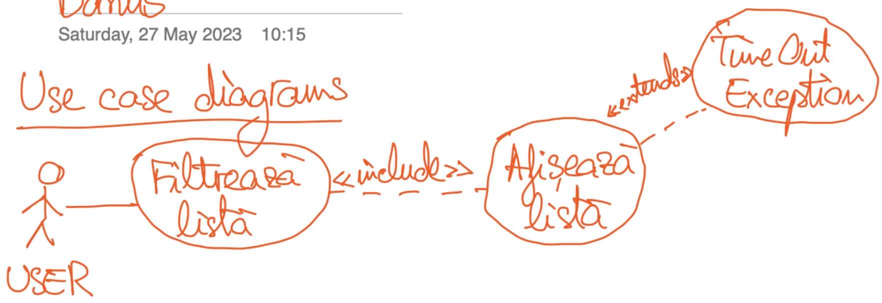
Bonus

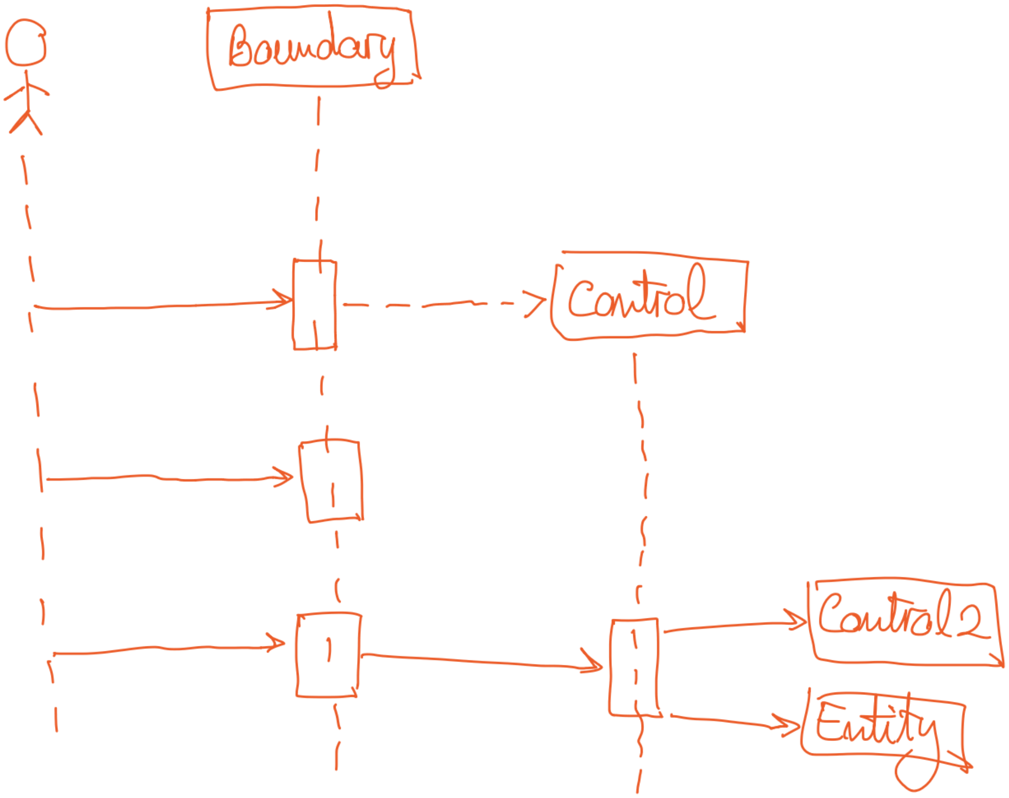
Saturday, 27 May 2023 10:15

Use case diagrams

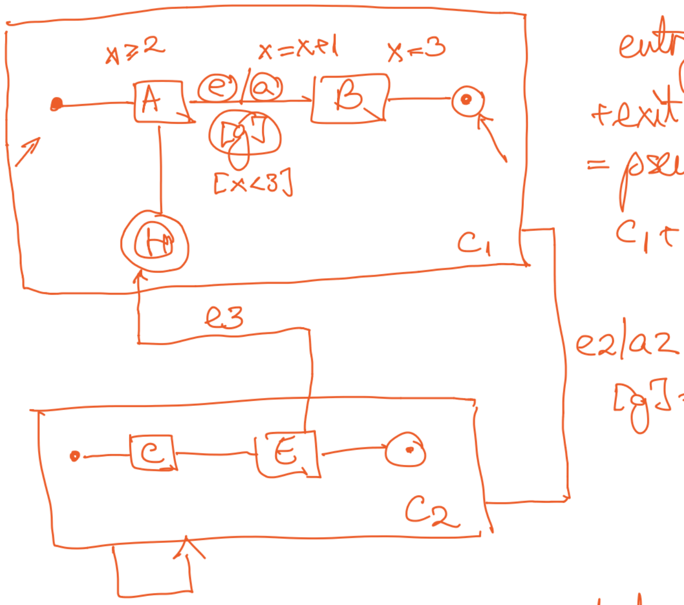


Sequence diagrams

1. Entity objects - objects persisting in the system
2. Boundary objects - UI/GUI objects (e.g. pointer, cursor)
3. Control objects



State Machine Diagrams



entry point + exit point = pseudo-states
+ history = (H)

C1 + C2 = composed states

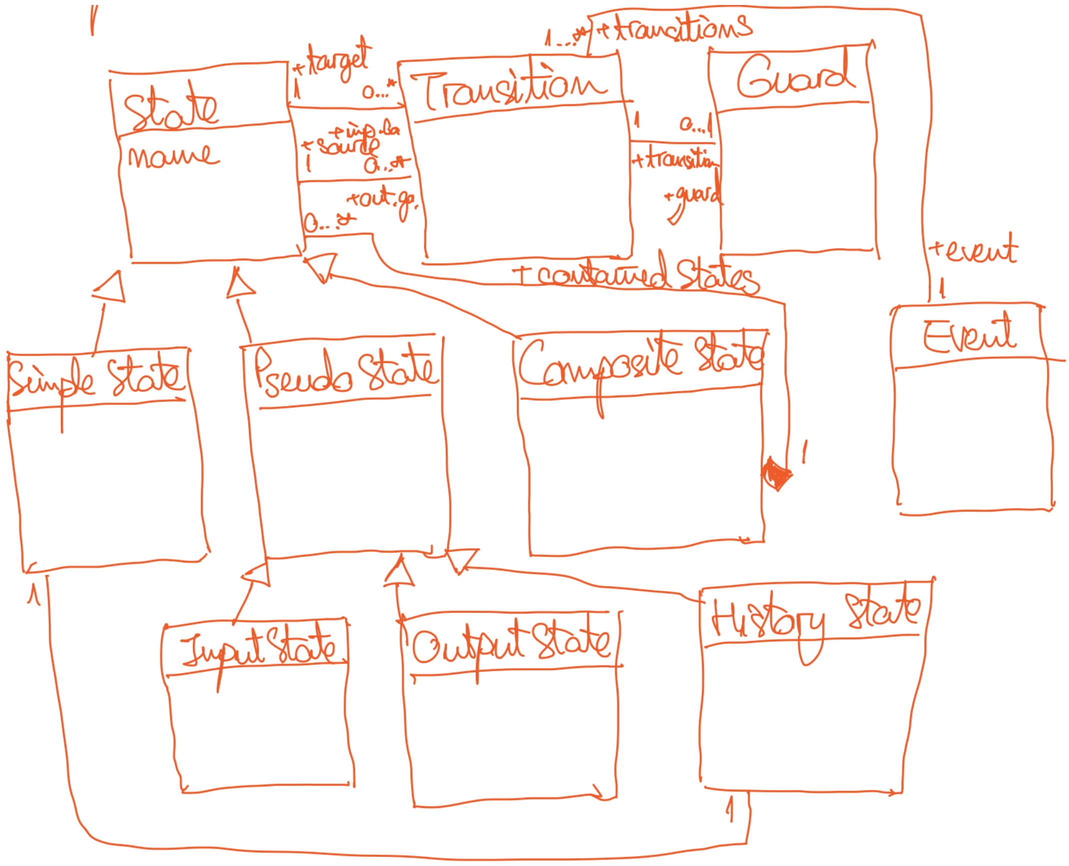
[g] = guard

- they separate the concerns between the entities, whereas seq. diag. show the interaction

- Boundary vs. Control ?
- What is testing and its importance?
- Importance of modelling

State, Transition, Event, Guard

Simple States, Composite States, Pseudo-States
Input states, Output States, History States



```
Transiton {
    State source;
    State target;
}
```

```
State {
    Set<Transition> outgoing;
    Set<Transition> incoming;
}
```

OCL

```
context OutputState:
    inv outputState inv:
        self.outgoing -> isEmpty() and not
        self.incoming -> isEmpty()
```

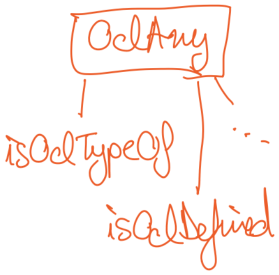
self.collection → collect[c | c.name]

syntactic sugar

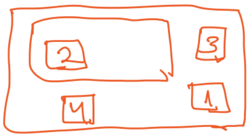
⇓

$\boxed{\text{self.collection.name}}$

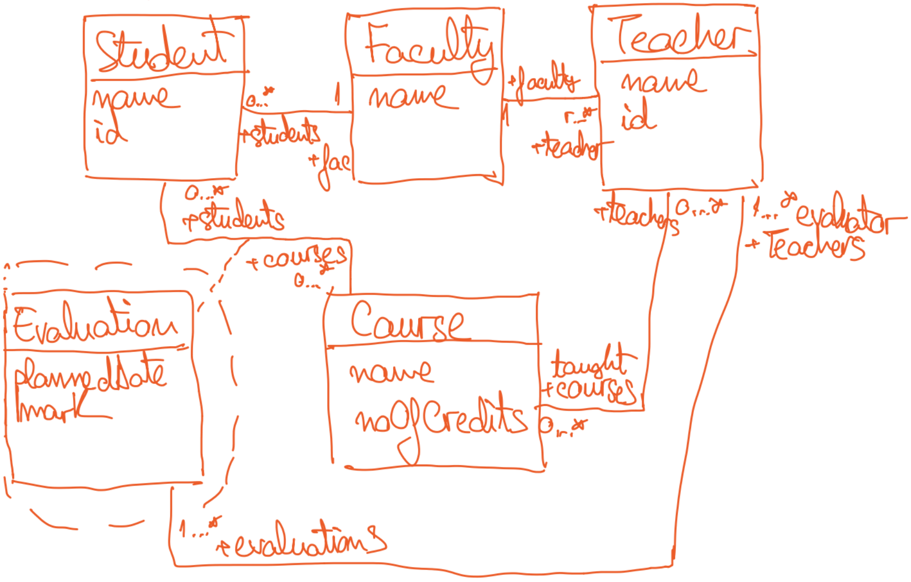
```
context InputState:
  inv inputStateInv:
    self.incoming → isEmpty() and
    self.outgoing → size() = 1
```



```
context CompositeState:
  def numOfSimpleStates:
    let numOfSimpleStates: Integer =
    =self.containedStates → select(s:state)
|s.isOclTypeOf(SimpleState)) → size()
      let numOfInnerSimpleStates: Integer =
=self.containedStates.flatten → select(s: State)
```



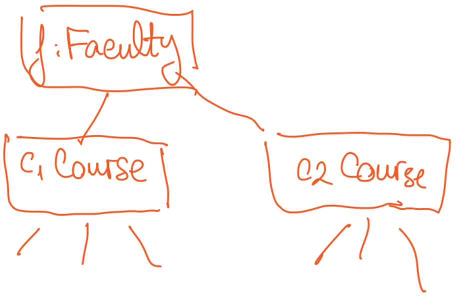
Faculty, Student, Teacher, Course, Evaluation



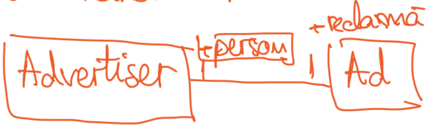
```
context Student:
  inv uniqueId :
     self.faculty.students.id->select(self.id) ->
  -> size() = 1

context Evaluation:
  inv associatedTeachers :
    self.course.teachers->includesAll(self.evaluatorTeachers

context Student:
  def passedEvals:
     let passedStudent: Boolean = self.courses ->
   select(c c | c.evaluatio.mark < 5 and
   s.evaluation.student = self) -> size() = 0
```

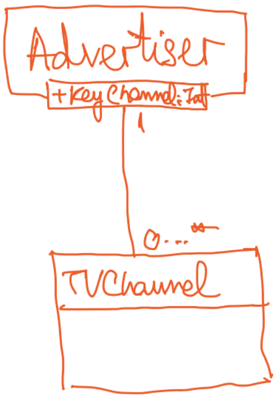


addParents, removeParents / ~~add Parent~~



```
Advertiser {                 Ad {
  Ad ~~ad~~; reclama;           Advertiser person;
}                            }
```

⇒ if the field is specified in the code, you MUST find it in the UML!!



Map TVChannel;

- Software Dev Life Cycles ?

(req. elicitation)