Notes for Proj#2

Sequence diagrams, Activity diagrams and State-chart diagrams

State-chart diagram

- Must have “start” and “final” nodes
- Need “event definition/description”
  - “Export confirmed”, “Temperature exceeds the threshold”, ...
- Transitions (arrows) with events
- ~~Name of state: noun phrase~~
- Name of state: adjectives
  - “Waiting”, “Pumping”, “Accepted”, ...
  - Unique!
- Actions in each state
  - enter: perform the actions when entering the state
  - exit: perform the actions before exiting the state
  - do: perform the actions while in the state

State-chart diagram

- NOT “screen flow” / “menu chart” !

Activity diagram

- Must have “start” and “final” nodes
- Notation: activity = rounded rectangle
  - Is an action
- Can use swimming lanes
- Use decision node (diamond) for branching
  - Branching conditions are on out arrows
  - Nothing inside the diamond (NOT a flow chart)
  - (Can have an activity before the decision for making the branching conditions)
- NOT decision node for joining the branches!

Activity diagram

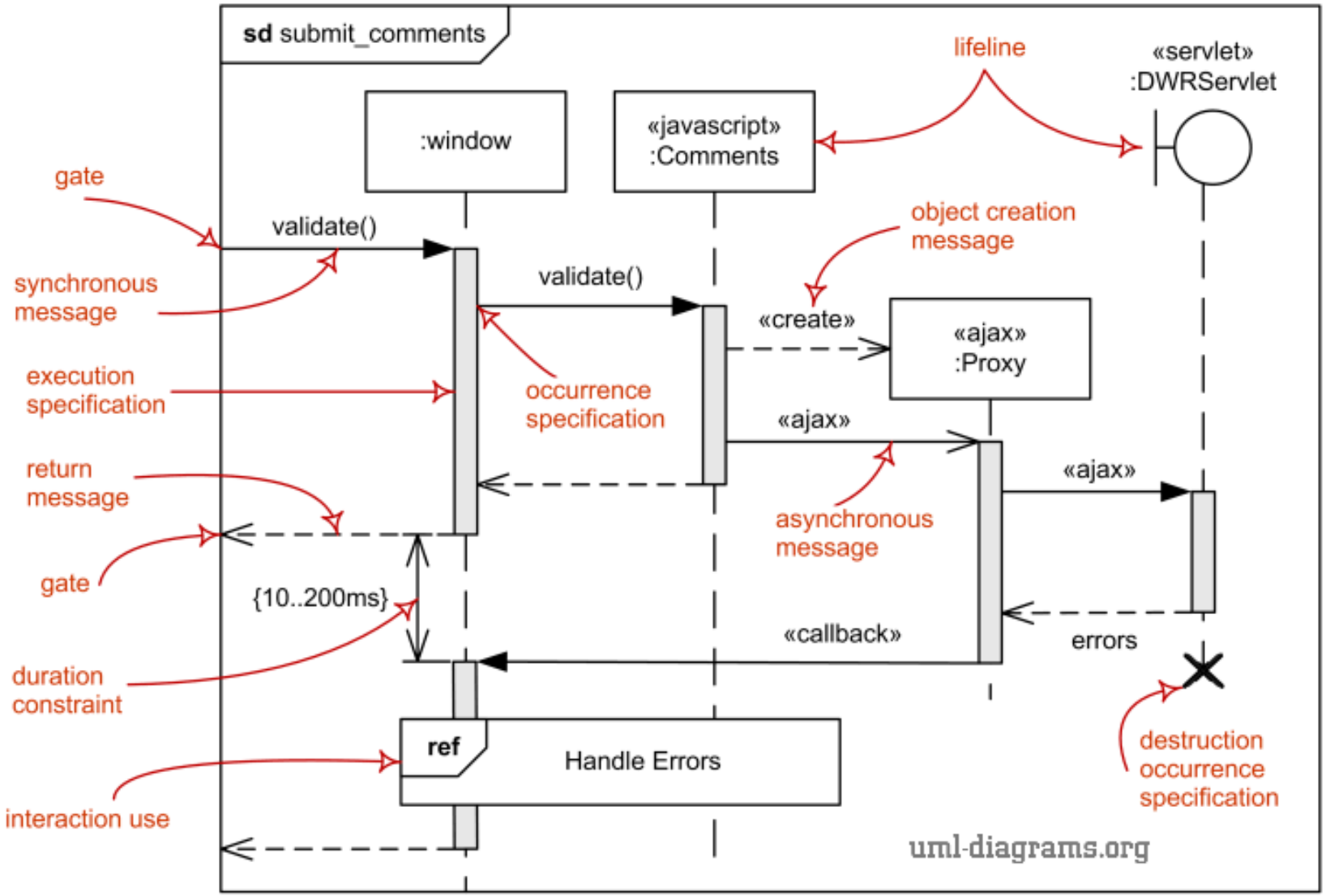
- Use fork node (parallel = a horizontal or vertical bar) for concurrent activities
  - Decision node: only one branch may happen
- Use join node (a horizontal or vertical bar) for joining all concurrent activity
- NO join node for joining the decision branches!

Sequence diagram

- Notation:
  - Object/Component:
    - Simple: noun
    - Correct: “object:Class”
  - Solid filed arrow: an interaction (ask for an operation/call a method)
  - Dashed open arrow: result of an operation/method
    - Name of the result, noun (NO “Return the information”)
  - Long rectangle: the activation of an interaction/lifetime of an operation/execution time of a method
    - When A --> B --> C then C ends before the end of B, and B ends before the end of A
    - Except “asynchronous operation”
  - Solid open arrow: an asynchronous openration
    - (May be together with a callback)

https://www.uml-diagrams.org/sequence-diagrams.html

Sequence diagram



Sequence diagram

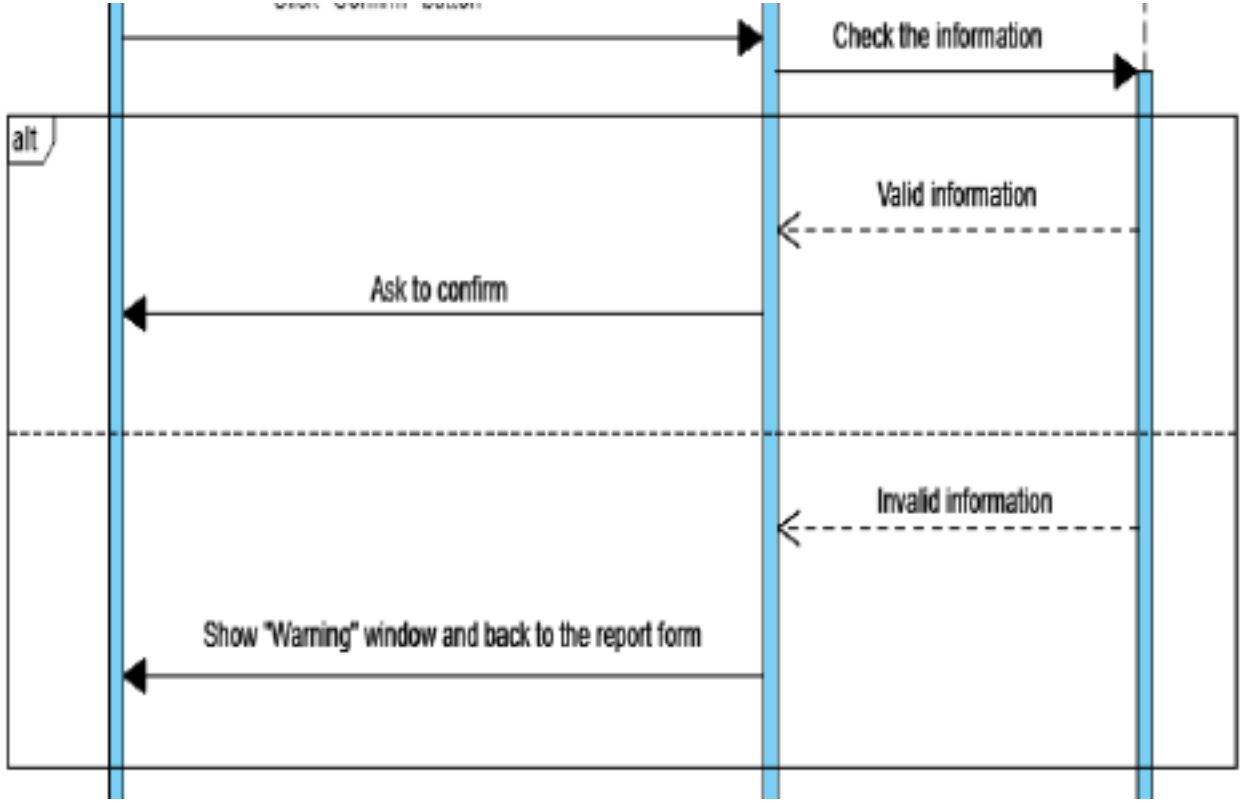
- Verb in the interaction?
  - NO: “Processing Unit” ----“send data to server”----> “Server”
    - “send...” is defined and performed in “Processing Unit”
  - Correct: A ----“operation”----> B: “operation” is defined/belong to B
- Meaning interactions (and different from other interactions)
  - NO: “get()” – almost no meaning
- NO “orphan operations”
  - Have to be activated/triggered by actors or other operations
- Order in time

Sequence diagram

- alternative: “alt” (if-then-else/switch)
  - If the branching conditions use information, the information may have to be returned from previous operation and BEFORE and OUTSIDE the “alt”
  - Branching conditions may totally different
- Optional: “opt” (if without else)
  - Only one case

Sequence diagram

- Example: may change as:
  - Result of the “Check the information” as “check result”
  - Condition 1
    - [check result = ok]
  - Condition 2
    - [check result = invalid]



Sequence diagram

- NO NEED for each use-case
  - A sequence diagram may be for one or more use-case
    - For more use-case? Don’t make it complicated, kills the readers!
  - Sequence diagram for no use-case?
    - Maybe: just for internal interaction between internal components!
- Sequence diagram = internal interaction
  - NEED internal components!
  - And they interact with each others!
- Actors in use-cases = actors in sequence diagrams!

- State diagram and activity diagram for use-case?
  - NOT for use-case!!!
- State diagram vs. activity diagram
  - State diagram: choose an object for clarifying its states and the transitions of its states
  - Activity diagram: choose a problem to be presented (algorithm for example)
- This diagram for what?
  - Name
  - For? (functionality, object, …)