UML Behavioural Modelling

HUSZERL Gábor
huszerl@mit.bme.hu

MŰEGYETEM 1782
BME VILLAMOSMÉRNÖKI ÉS INFORMATIKAI KAR
Méréstechnika és Információs Rendszerek Tanszék
ftsrg Critical Systems Research Group

Learning Outcomes

- At the end of the lecture the students are expected to be able to

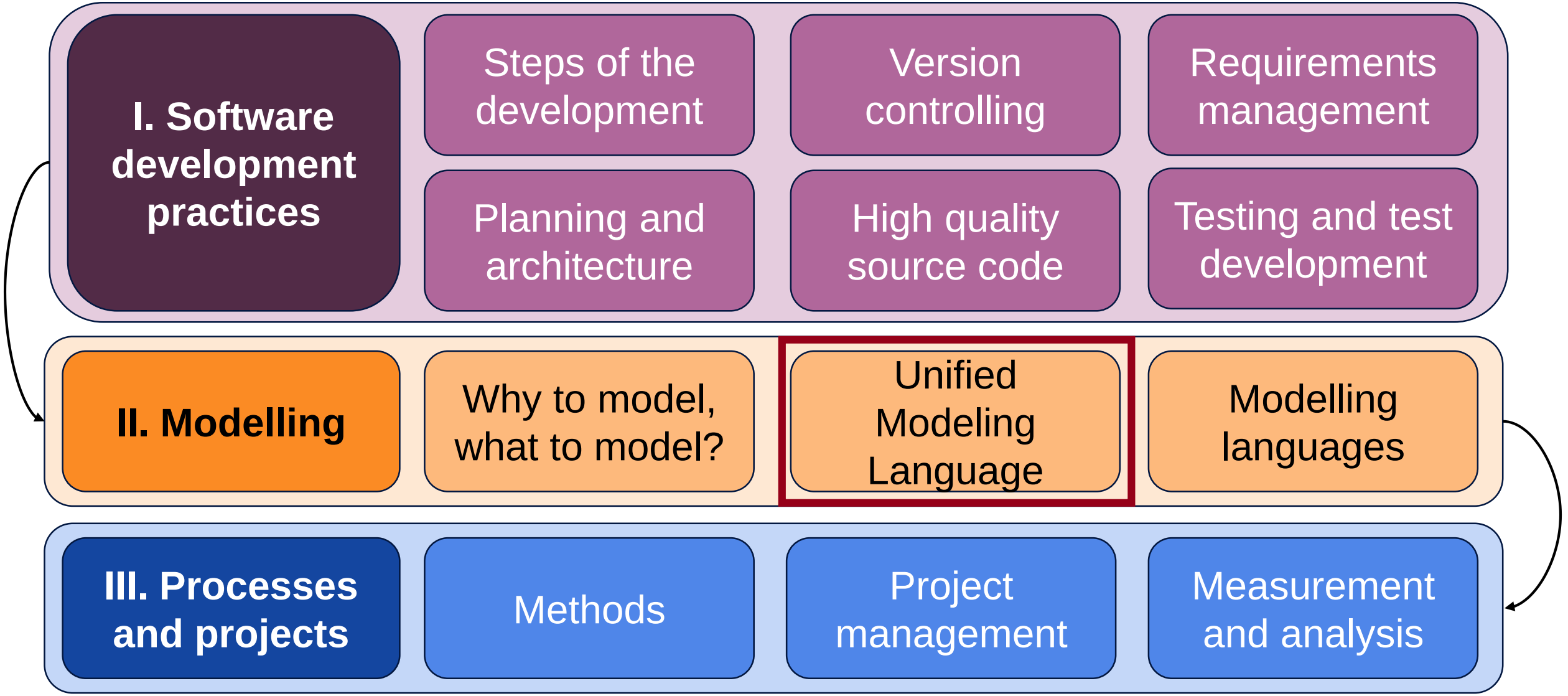
- (K1) recall the objectives of structural modelling in UML,

- (K3) use the UML activity diagrams to model processes,

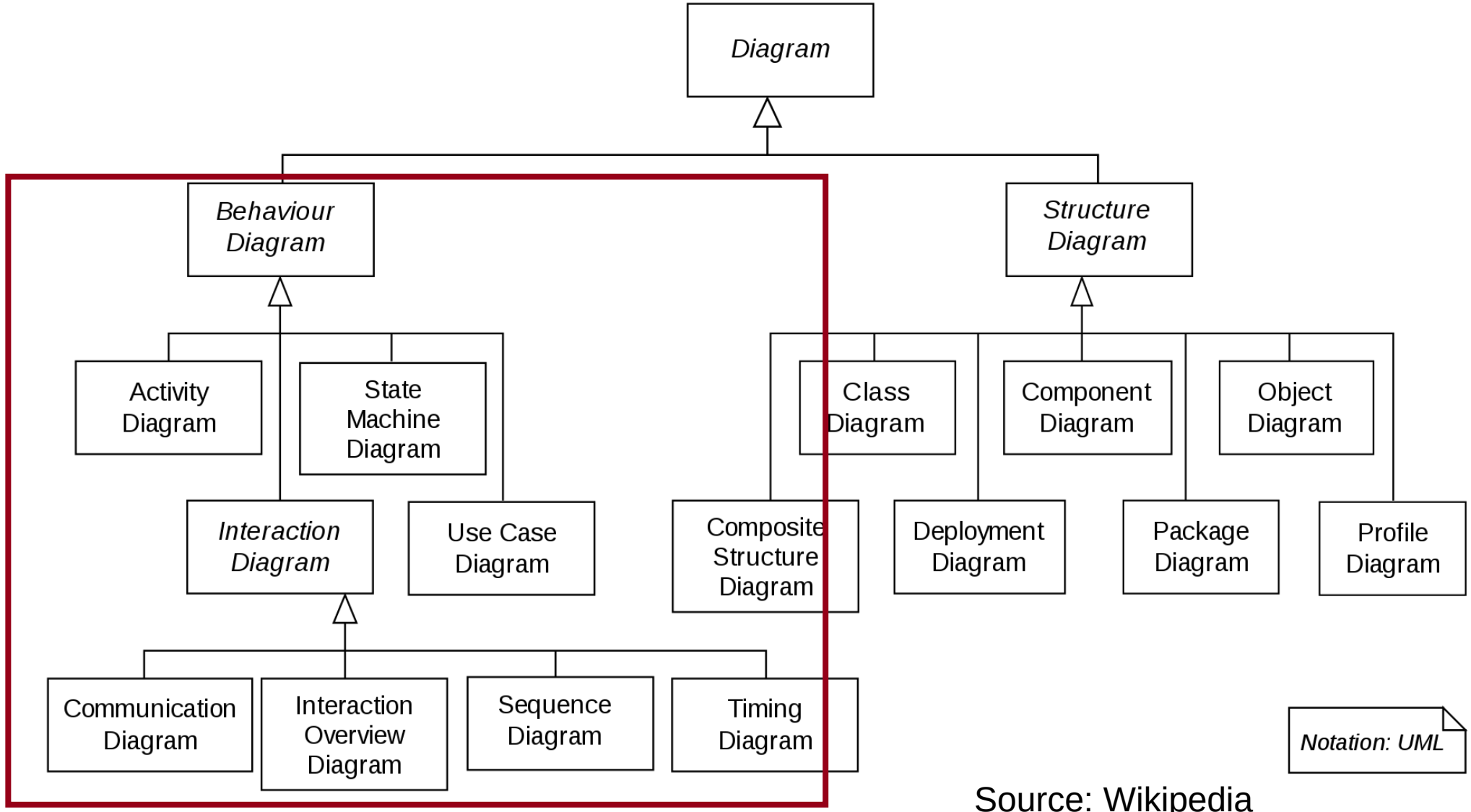
- (K3) use the UML state machines to model reactive systems,

- (K3) use the UML interactions to model scenarios.

Further Topics of the Subject



UML Diagram Types



Source: Wikipedia

Aspects of Modelling Behaviour

- What can the system do?
  - How does the state of the system change over time?
  - What are the expected and/or prohibited behaviours?
  - How does the system react to the changes in its environment?
  - In which order should the steps be executed?

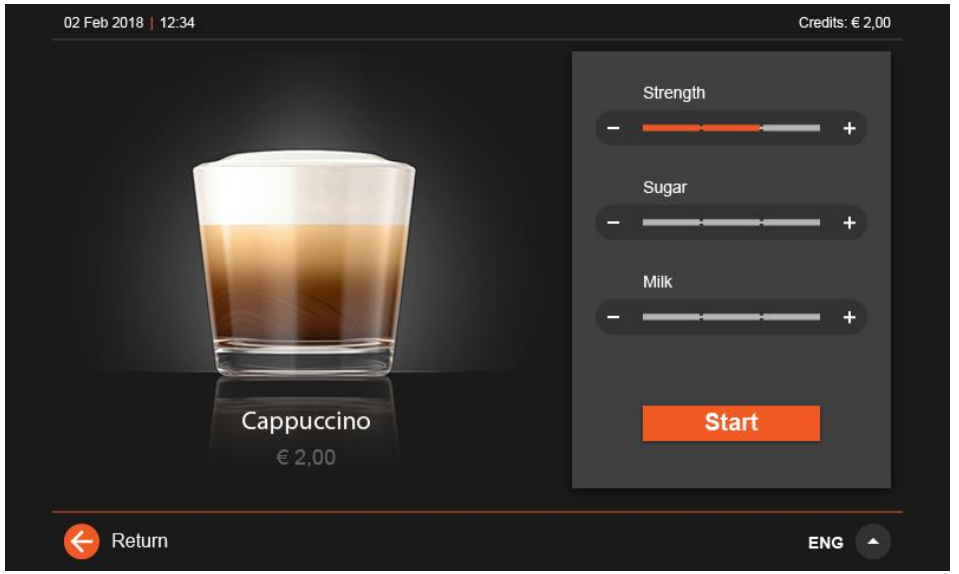
- **Main approaches**
  - State based models
  - Control flow models
  - Data flow models

What Behaviour Can Be Modelled?

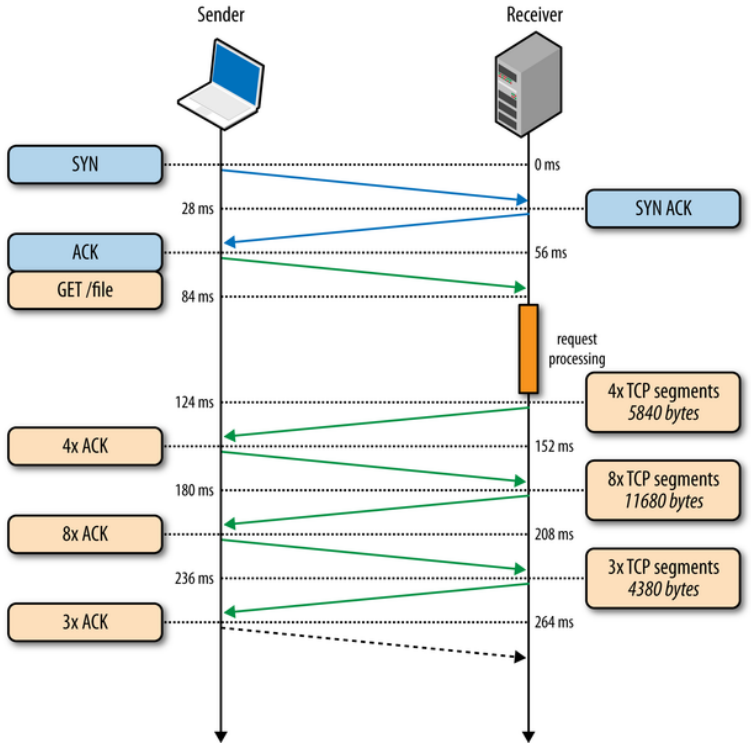
Internal algorithms

```
private void analyzeRoutes()
{
        for (String routeName : routeNames) {
                Collection<Route> versions = nameToRoute.get(routeName);
                List<Trip> trips = new ArrayList<>();
                for (Route route : versions) {
                        trips.addAll(routeIdToTrips.get(route.getId()));
                }

                if (trips.isEmpty()) {
                        System.out.println(
                                        String.format("%s: no trips found", routeN
                        continue;
                }

                System.out.println(
                                String.format("%s: %d trips", routeName, trips.siz
                Multiset<StopIdList> stopIdListSet = HashMultiset.create();
                for (Trip trip : trips) {
                        StopIdList stopsIds = tripIdToStopList.get(trip.getId());
                        stopIdListSet.add(stopsIds);
                }

                Multiset<StopIdList> histogram = Multisets
                                .copyHighestCountFirst(stopIdListSet);

                StopIdList longest = null;
                int maxStops = -1;
                for (StopIdList stopIds : histogram.elementSet()) {
                        if (stopIds.size() > maxStops) {
                                maxStops = stopIds.size();
                                longest = stopIds;
                        }
```

Behaviour and reactions of a system



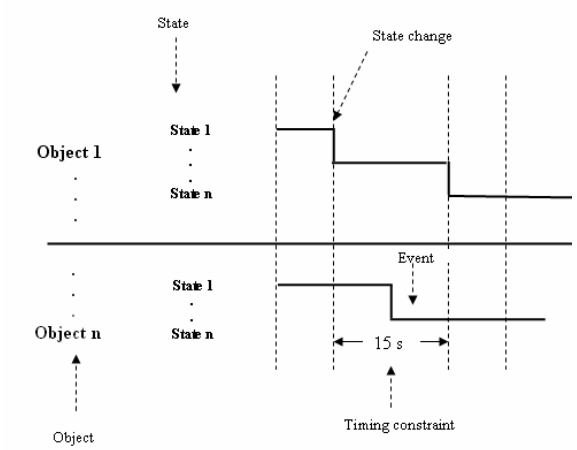
Protocol among multiple parties



What Can UML Model and What It Cannot

It can model

Discrete time/value behaviour



Characteristics:

- Discrete, typically finite state space
- Instantaneous transitions
- Reactions to events (invocation, input, …)

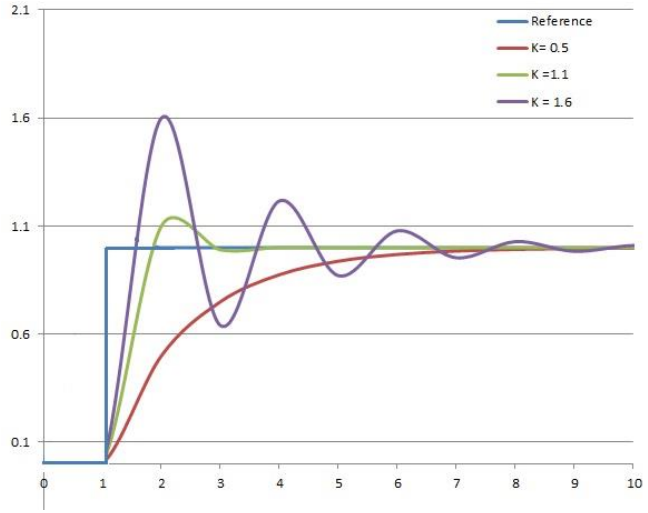
It cannot model (well)

Continuous time/value behaviour

$$\frac{dy}{dx} = f(x)$$

$$\frac{dy}{dx} = f(x, y)$$

$$x_1 \frac{\partial y}{\partial x_1} + x_2 \frac{\partial y}{\partial x_2} = y$$



Modelling:

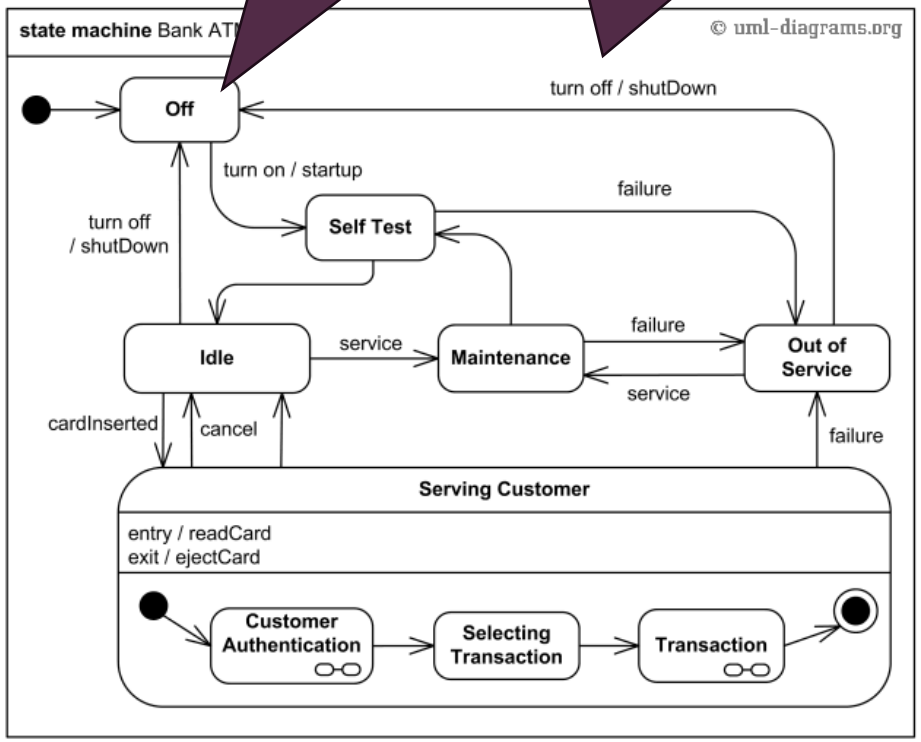
- Differential equations, controllers, …
- MATLAB/Simulink, Modelica, ...

See the course **System Theory**

UML Behavioural Models (1)

Discrete states of the system (~noun), waits for events

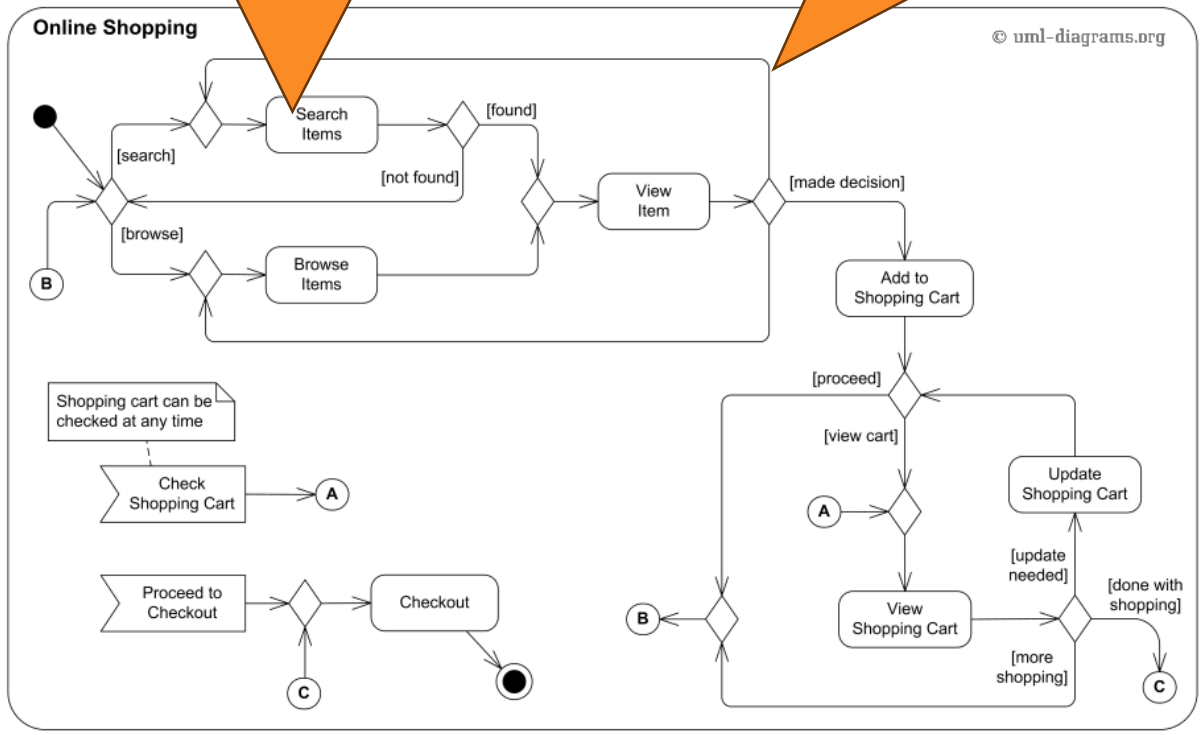
Response to external events and change of state



State machine

Steps: actions (~verb) Elementary behaviour, may change the state

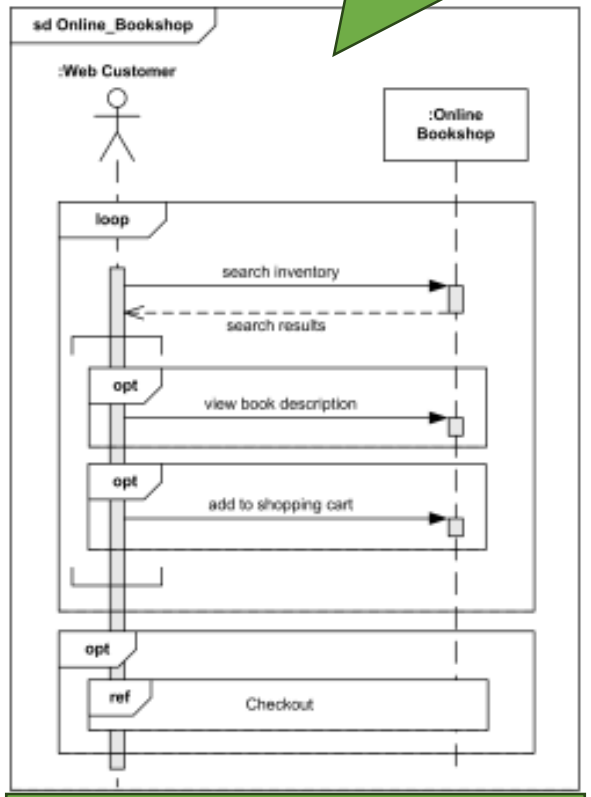
Edges: specifying dependency and ordering



Activity

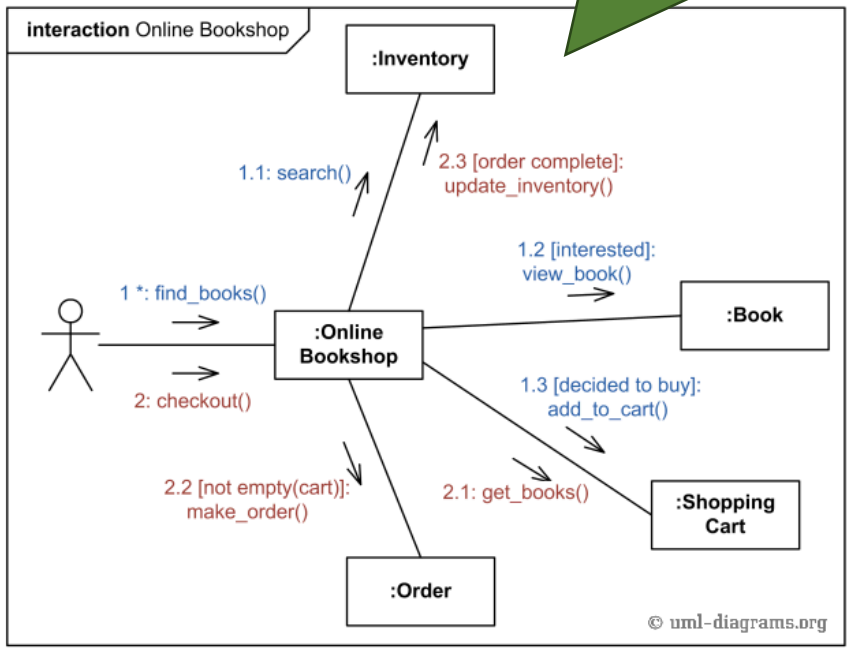
UML Behavioural Models (2)

Representation of one/more possible trace(s)



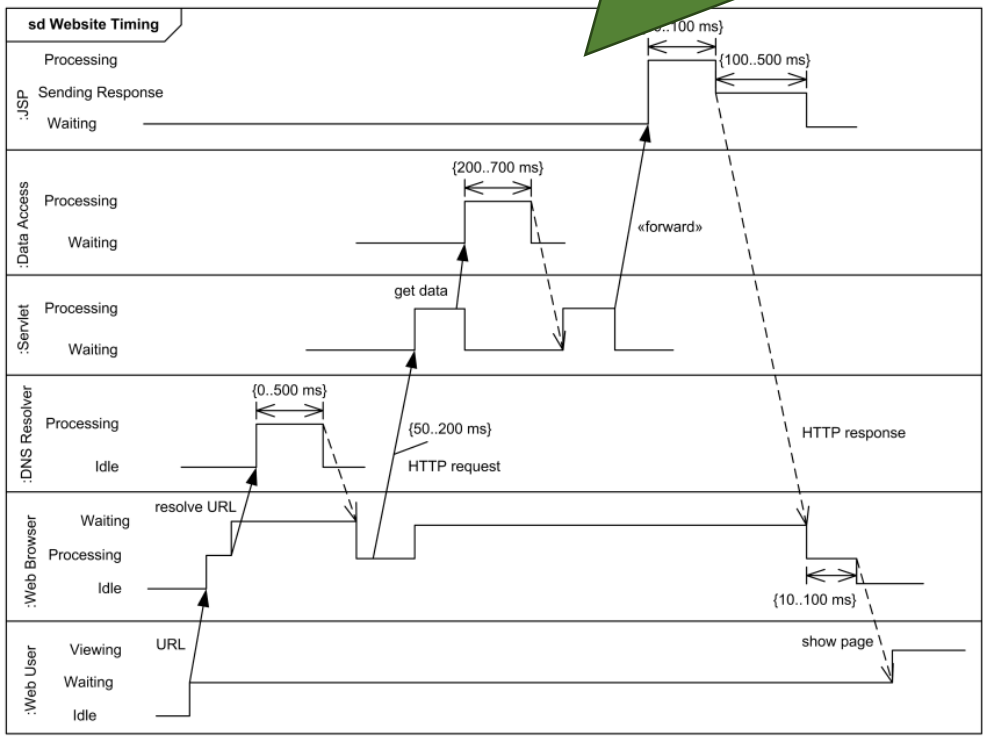
Sequence diagram

Representation of a simple trace



Communication diagram

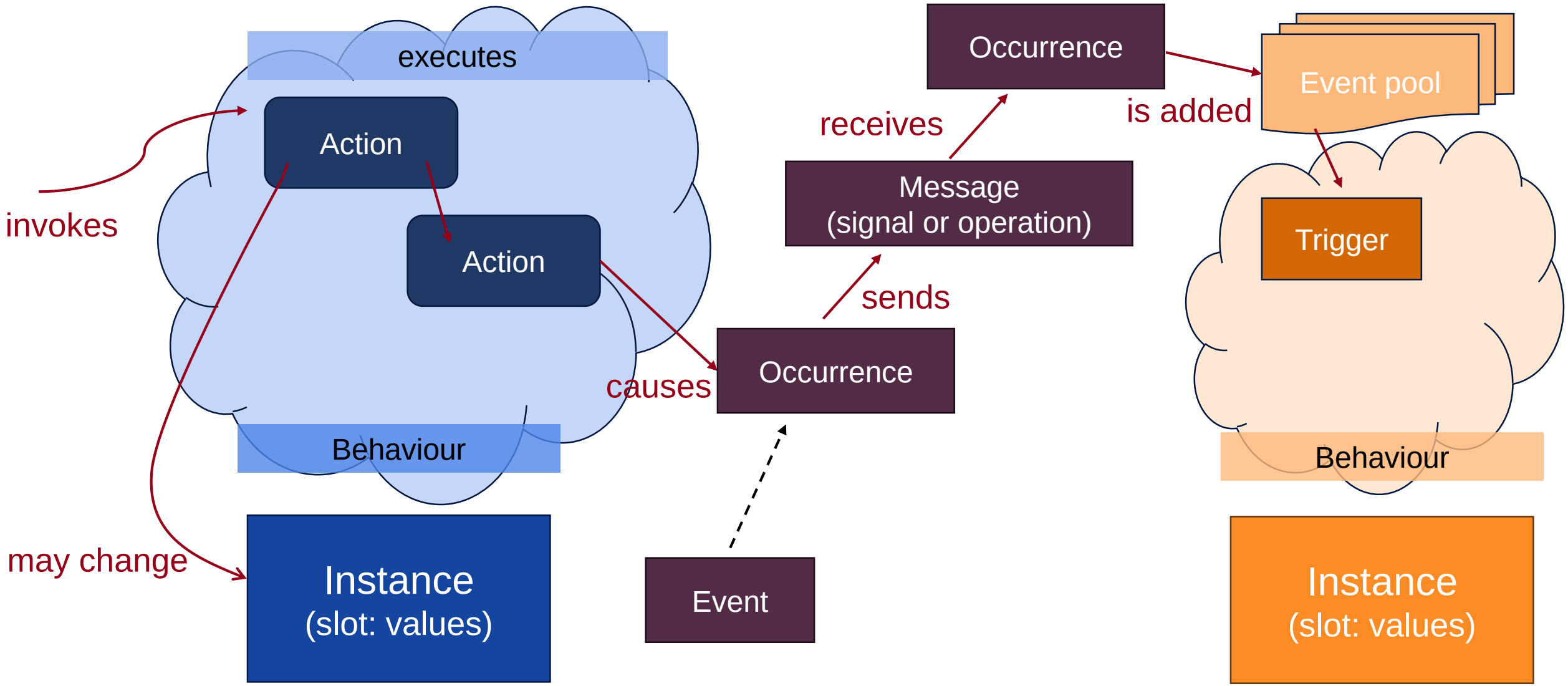
Representation of timing: both on internal and external changes



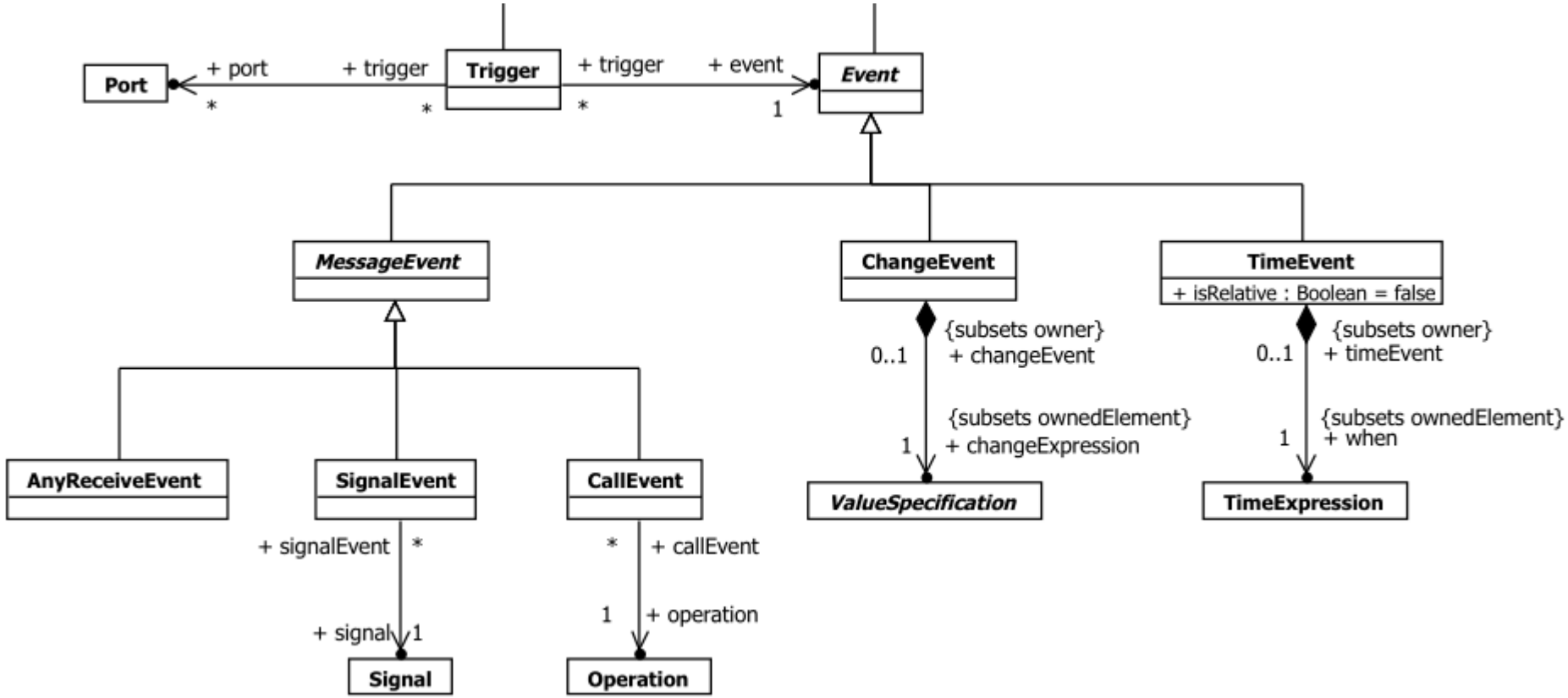
Timing diagram

Each of them describe Interactions!

Common Behavioural Elements (Illustration!)



Events in UML



Active Object

- Acts "autonomously", answers calls/messages
- After creation, it starts its behaviour (classifierBehavior)
- Can receive signals (Reception)
- Notation:

Not only 1 active object can be in the system! (competition problems!)