UML introduction

Software Requirements and Design
T-216-GHOH

Skúli Arnlaugsson | 2. september 2019

HÁSKÓLINN Í REYKJAVÍK
REYKJAVÍK UNIVERSITY

What is UML

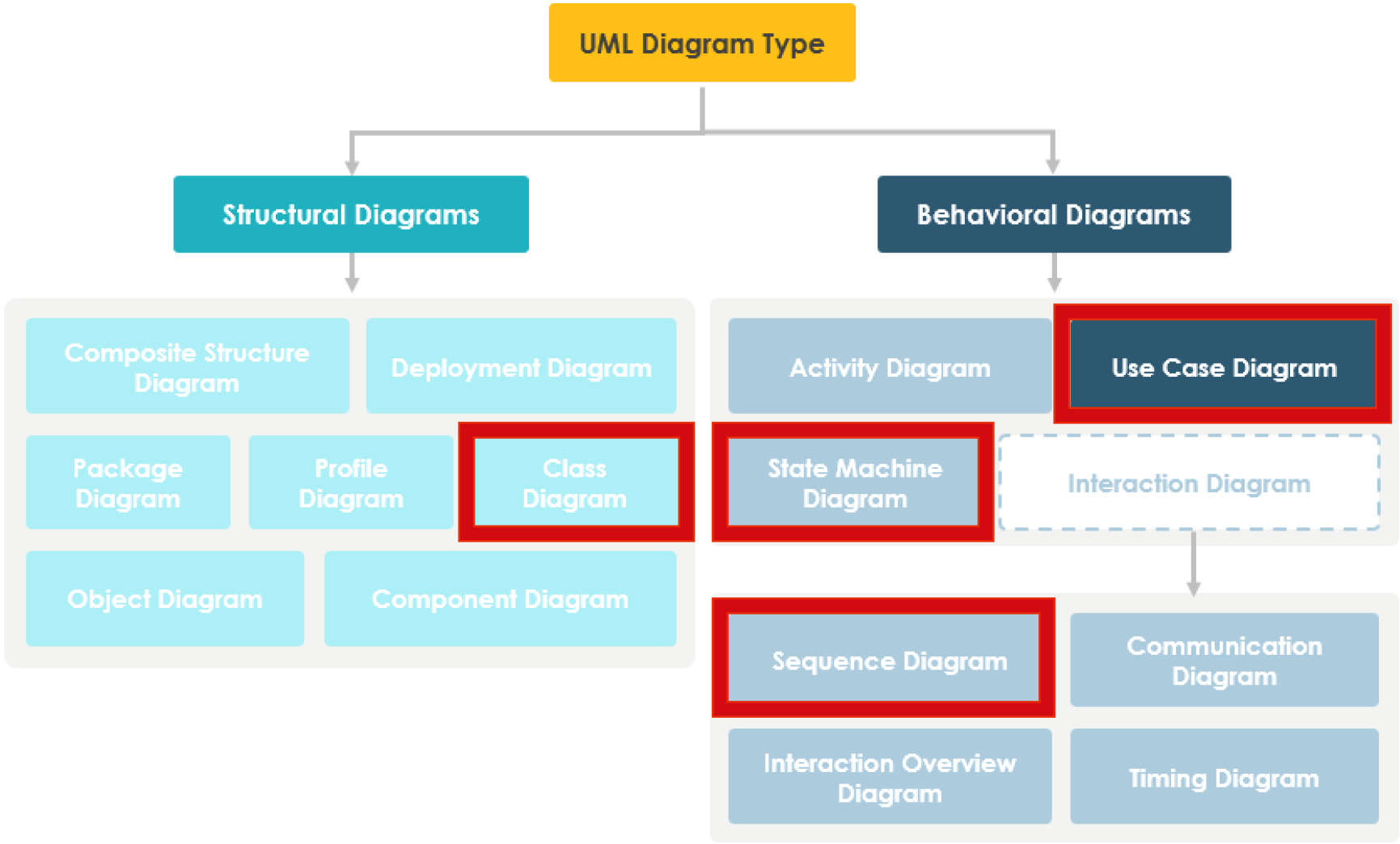
- A language used to *describe the behavior, functionality, and flow within a system* as well as *the system's interaction with the outside world*.
- A Modeling Language (mostly graphical)
  - Concepts (í. hugtök)
  - Notation (í. skrifháttur)
  - Organization (í. skipulag)

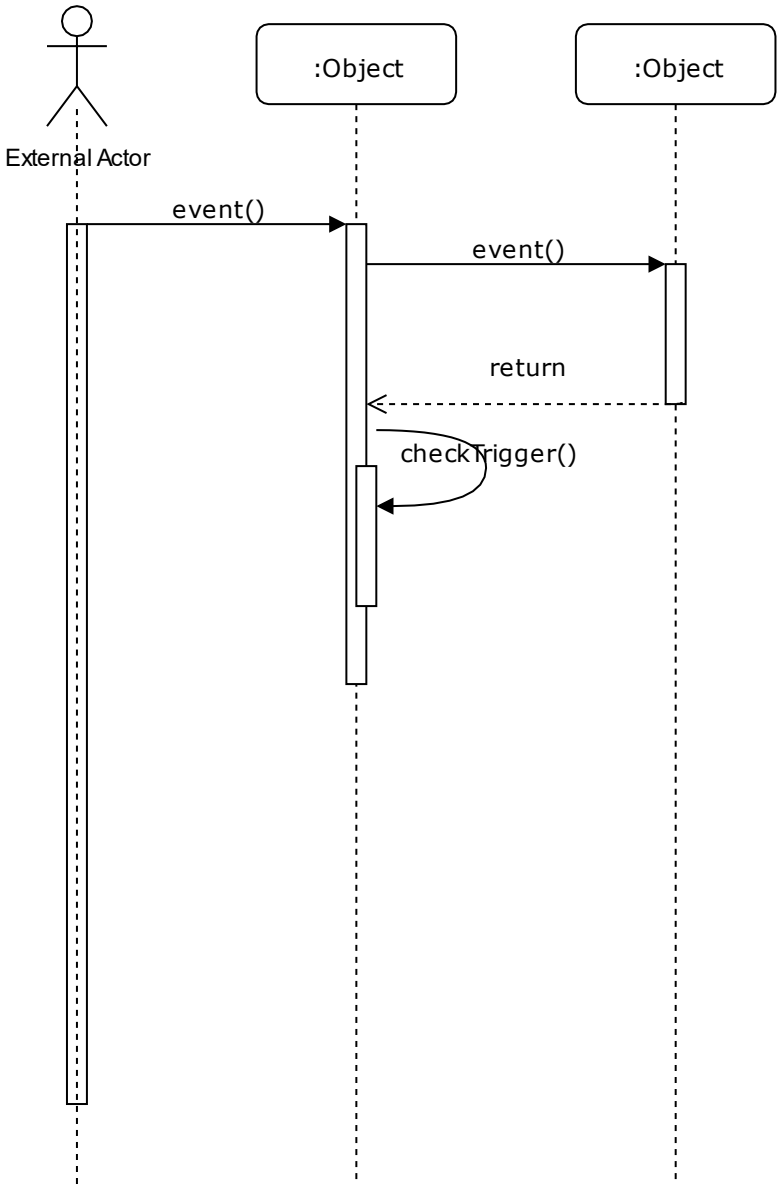
What UML is (cont.)

- *De facto* standard for modeling Software Systems
  - But not limited to only Software Systems
- Helps create clarity and understanding

What UML is (cont.)

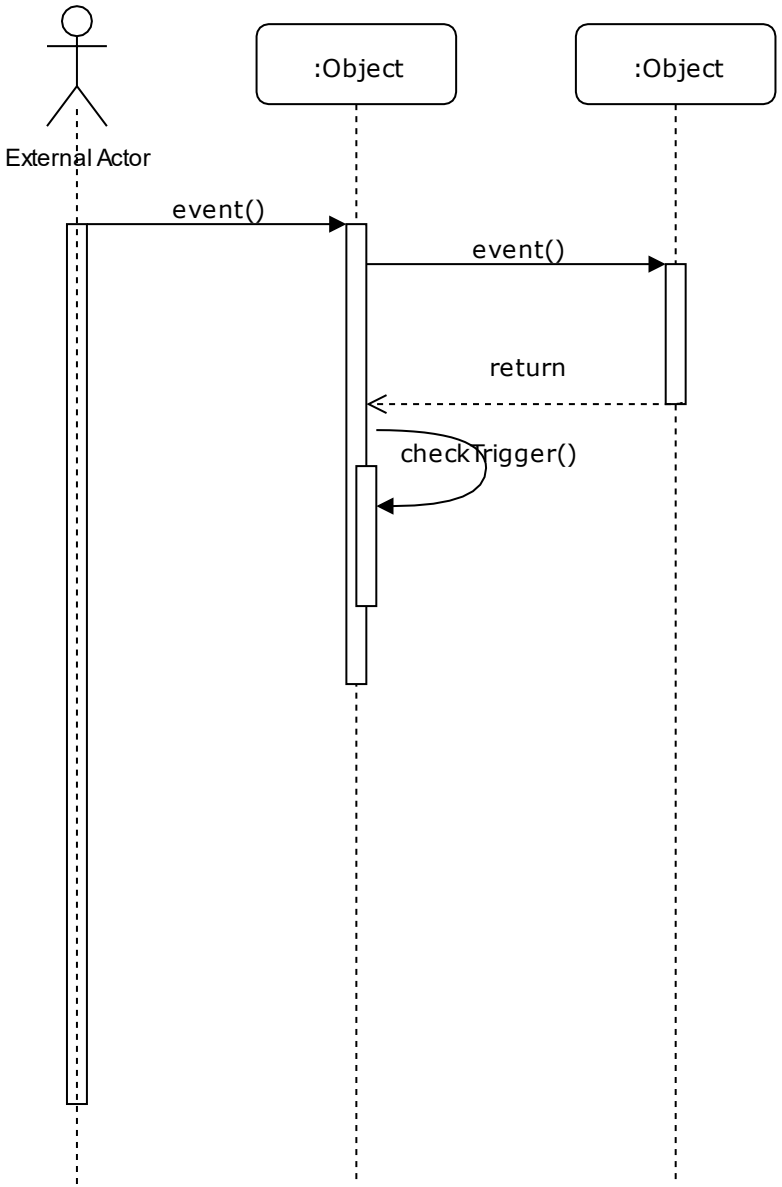
- UML is big! 800+ pages of specification
- In this course we will learn about a small subset of the UML standard
- Including:
  - Use cases and partly Use Case Diagrams (í. notkunartilvik)
  - Class Diagrams (í. klasarit)
  - State Diagrams (í. stöðurit)
  - Sequence Diagrams (í. runurit)


UML Diagram Type
Structural Diagrams
Behavioral Diagrams
Composite Structure Diagram
Deployment Diagram
Package Diagram
Profile Diagram
Class Diagram
Object Diagram
Component Diagram
Activity Diagram
Use Case Diagram
State Machine Diagram
Interaction Diagram
Sequence Diagram
Communication Diagram
Interaction Overview Diagram
Timing Diagram



Sequence Diagrams (í. runurit)

- Interaction Sequence Diagram
- Provide a view of the Interaction between Entities of the Systems
- Supports Use Cases
- Models a Sequence of actions and interactions

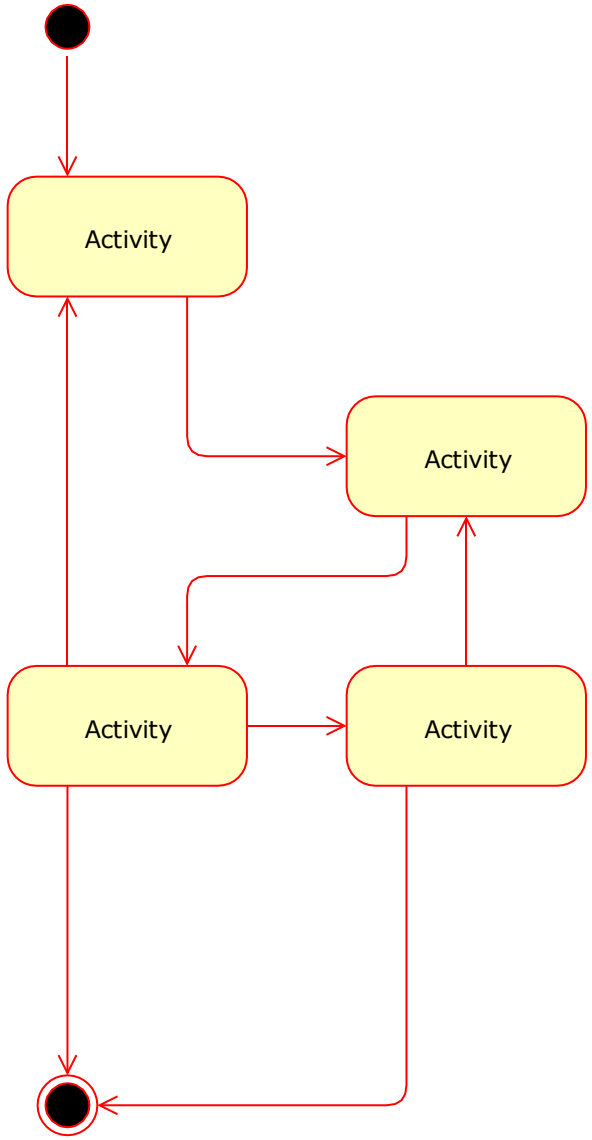


Sequence Diagrams (cont.)

- Concepts:
  - Actors: Roles or Participants
  - Activation or Execution Occurrence
  - Messages
  - Lifelines
- Notation:
  - How the diagram is set up
  - Meaning in symbols
    - F.x. Filled arrow head vs. empty arrowhead

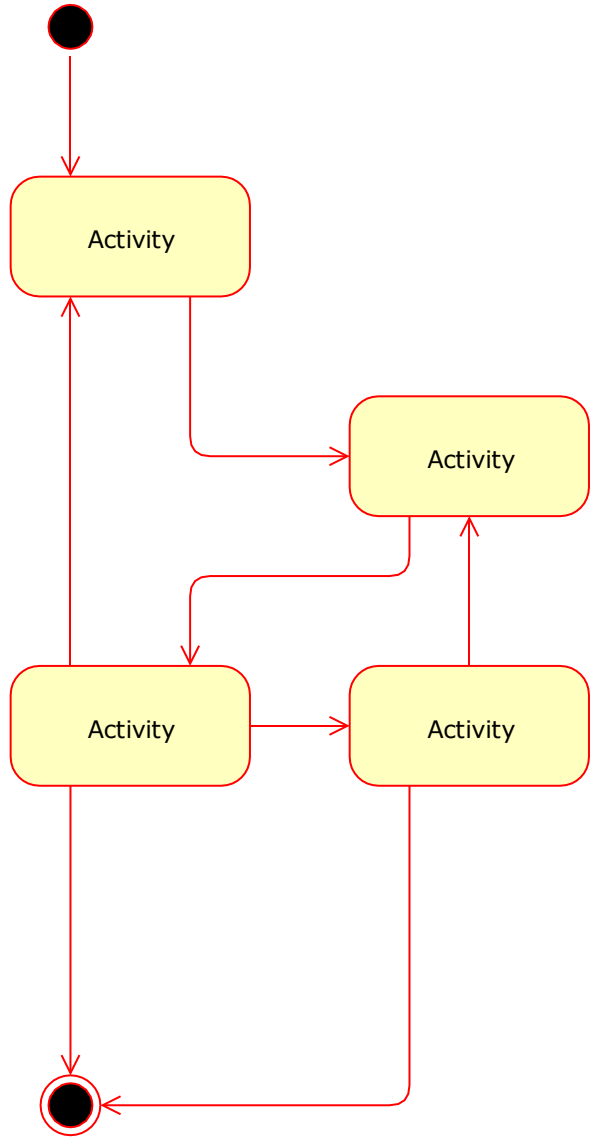
State Diagram (í. stöðurit)

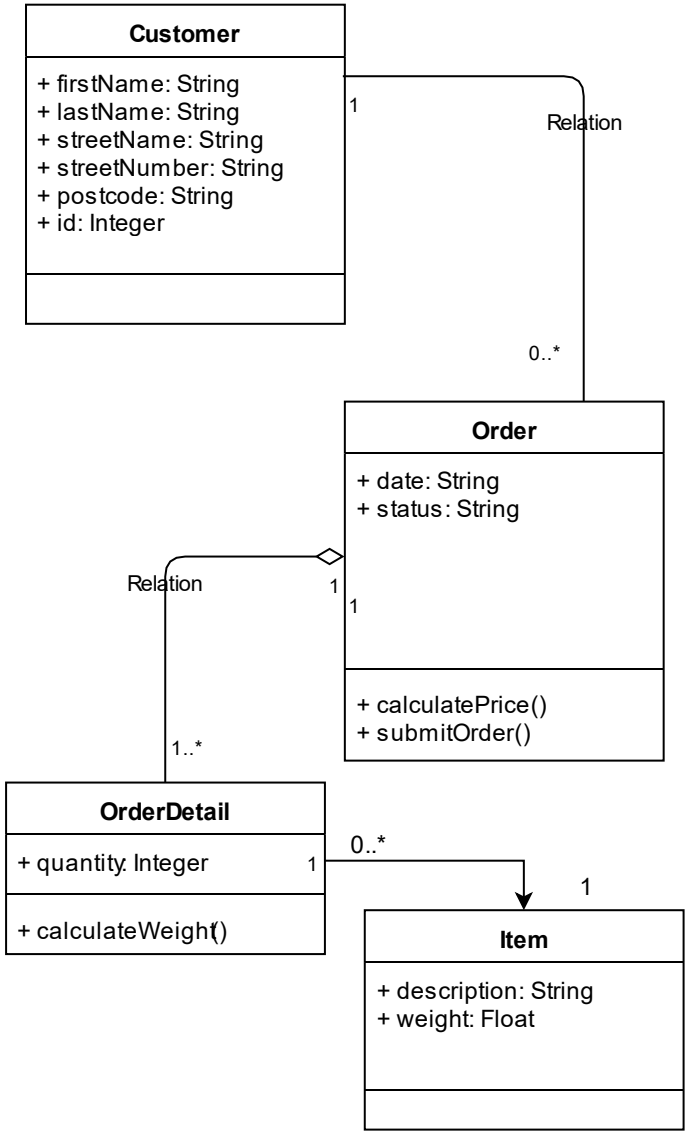
- State Machine Diagram
- Specify the behavior of some part of the system in terms of states and the transition between them
- Systems might behave differently depending on what state they are in, the State Diagram helps explain this behavior



State Diagram (cont.)

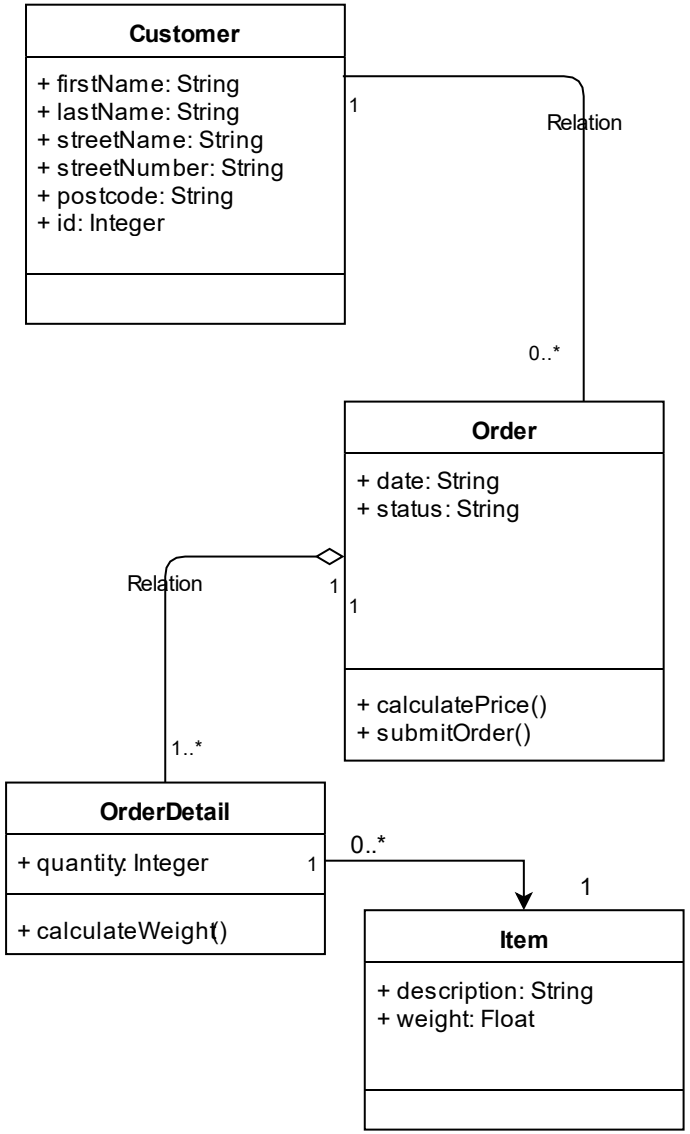
- Concepts:
  - States
  - Initial and final states
  - Transitions
    - Directed
    - Self-pointed transitions
    - Synchronization (join & fork)





Class Diagrams (í. klasarit)

- Model the Terms that are used in the System
- Model collaborations between Classes
- Reason about the System and how it will work
- Provide a basis for implementation in an Object Oriented Language



Class Diagrams (cont.)

- Concepts
  - Attributes
  - Behavior
  - Classes
  - Relationships
    - Inheritance
    - Association
    - Aggregation
    - Composition
    - Dependency

Wrapping up

- UML is not a Programming Language, but a Graphical Modeling Language that can be used to describe Systems and Behavior.
- The Goal for this Course isn’t to become Experts in UML, but to be *literate* when it comes to the most used Diagrams, and to **know how** and **when to use them**.


HÁSKÓLINN Í REYKJAVÍK
REYKJAVÍK UNIVERSITY