T E A C H I N G

D0020E PROJECT IN COMPUTER SCIENCE 2022/2023 LECTURE 7.1: MODELING

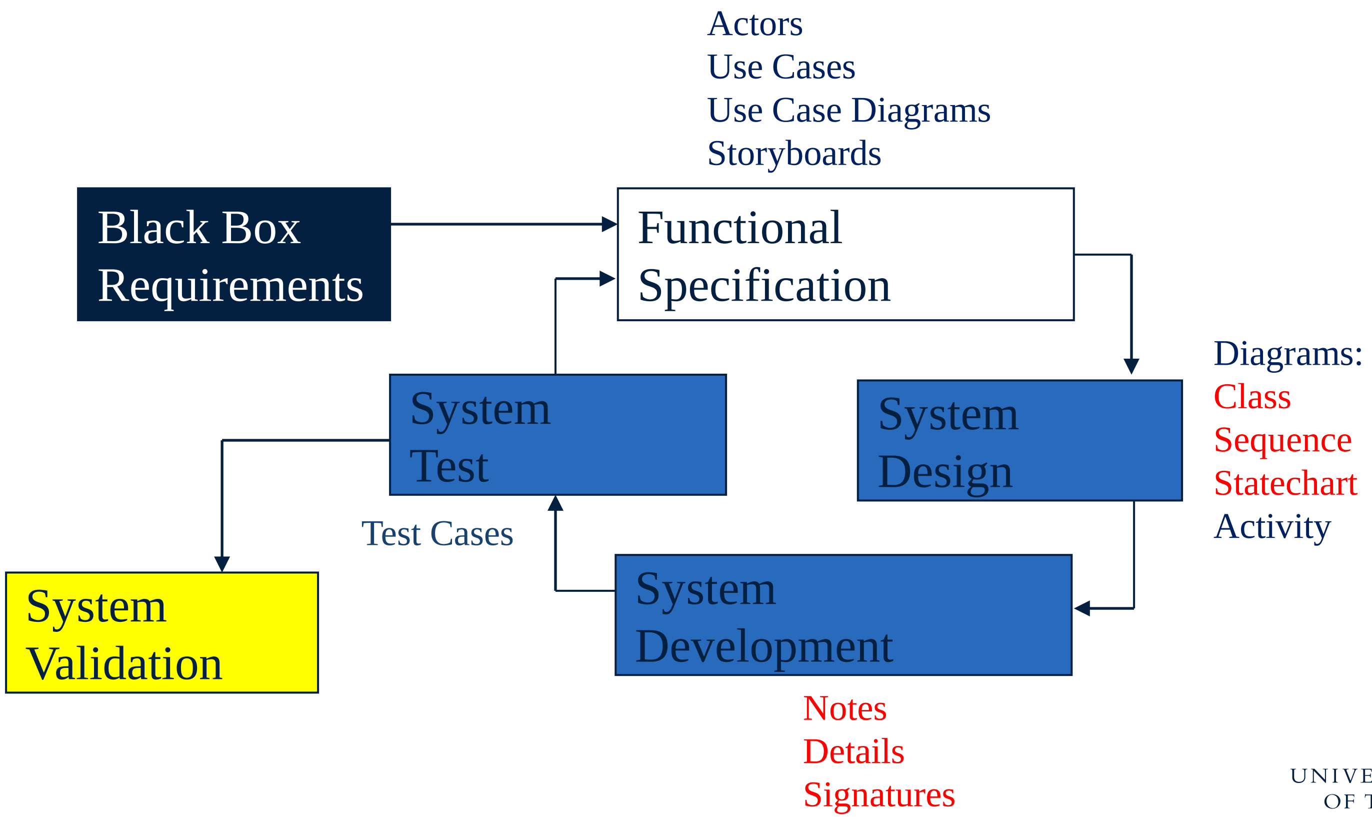
Ulf Bodin

LTU 23.11.2022

AS YOU CAN CLEARLY SEE IN SLIDE 397...
GAAAAH!
"POWERPOINT" POISONING.
www.dilbert.com scottadams@aol.com
© 2000 United Feature Syndicate, Inc.

LULEÅ UNIVERSITY OF TECHNOLOGY

Recap : UML artefacts



LULEÅ UNIVERSITY OF TECHNOLOGY

Today

- Modeling the run-time view!
- Software Design
  - Metrics
- Classification
- Code Review

LULEÅ UNIVERSITY OF TECHNOLOGY

Modeling using UML

Diagrams are the main artefacts of UML.
(These are not a sequence of steps!!!!)

- Use case diagrams
  - Functional behavior of the system as seen by the user.
- Activity diagrams
  - Dynamic behavior of a system expressed as a flowchart.
- Class diagrams
  - Static structure of the system: Objects, Attributes, and Associations.
- Sequence diagrams
  - Dynamic behavior between actors and system objects.
- Statechart diagrams
  - Dynamic behavior of an individual object.
- …

LULEÅ UNIVERSITY OF TECHNOLOGY

Use Case Example

Name: `Purchase ticket`

Participating actor: `Passenger`

Entry condition:

- `Passenger` standing in front of ticket distributor.
- `Passenger` has sufficient money to purchase ticket.

Exit condition:

- `Passenger` has ticket.

Event flow:

1. `Passenger` selects the number of zones to be traveled.
2. Distributor displays the amount due.
3. `Passenger` inserts money, of at least the amount due.
4. Distributor returns change.
5. Distributor issues ticket.

LULEÅ UNIVERSITY OF TECHNOLOGY

Sequence diagrams

- How do we design the flow for coders?
- What information do they need?

Example

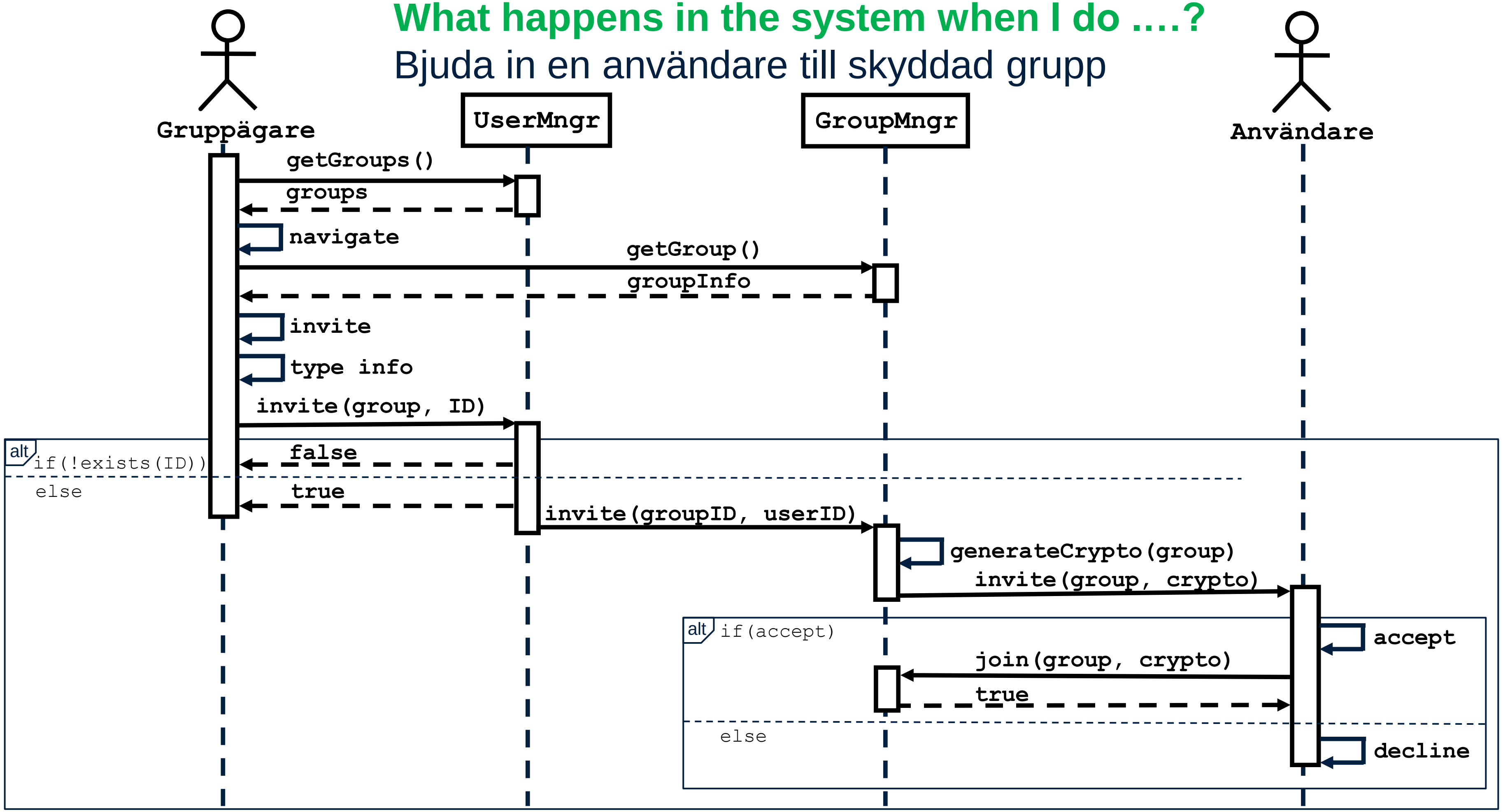
- A group communication app
  - Which categories of users exist?
  - What functions should exist?
  - How do I achieve ….?
  - What can I do with the app when I am at ….?
  - What happens in the system when I do ….?
  - How is …. Implemented?

Användarfallsbeskrivning

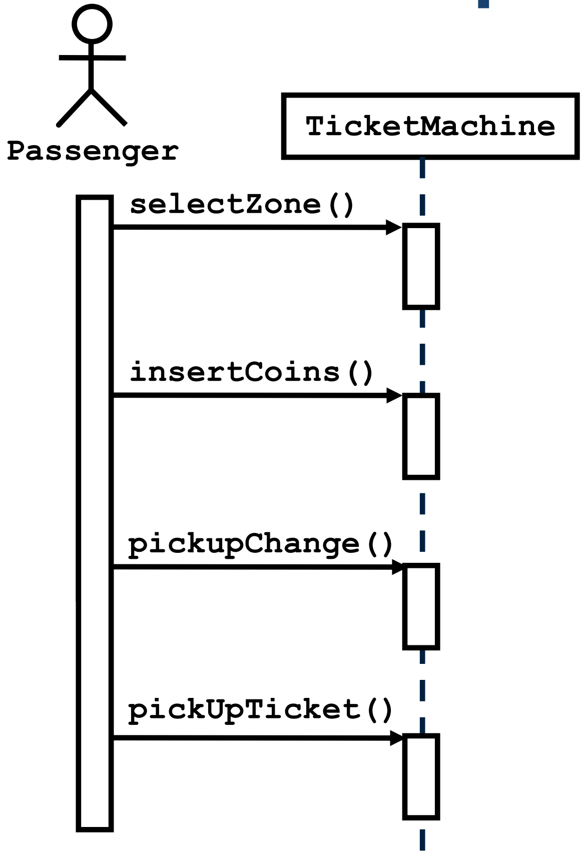
- Användarfall: Bjuda in en användare till skyddad grupp
  - Aktörer: Gruppägare, Användare
  - Start: Gruppägaren är inloggad och är på startvyn
  - Händelser:
    - Välj ”Groups”
    - Navigera till rätt grupp
    - Press ”Select”
    - Press “Options”
    - Press “Add/invite”
    - Skriv in kontaktinformation
    - Användaren får notis om inbjudan
      - Användaren väljer “accept”
  - Avslut: En ny användare har bjudits in till gruppen

What happens in the system when I do ….?

Bjuda in en användare till skyddad grupp

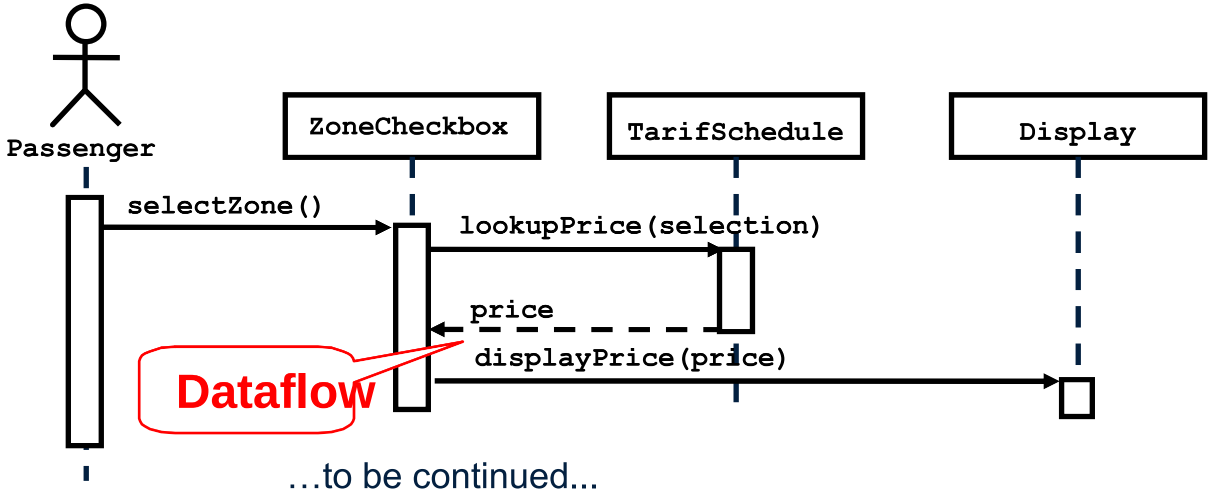


UML Sequence Diagrams



- In requirements analysis
  - To refine use case descriptions
  - to find additional objects (“participating objects”)
- In system design
  - to refine subsystem interfaces
- Columns = classes
- Arrows = messages
- Narrow rectangles = *activations*
- Dashed lines = lifelines

UML Sequence Diagrams: Nested Messages



…to be continued...

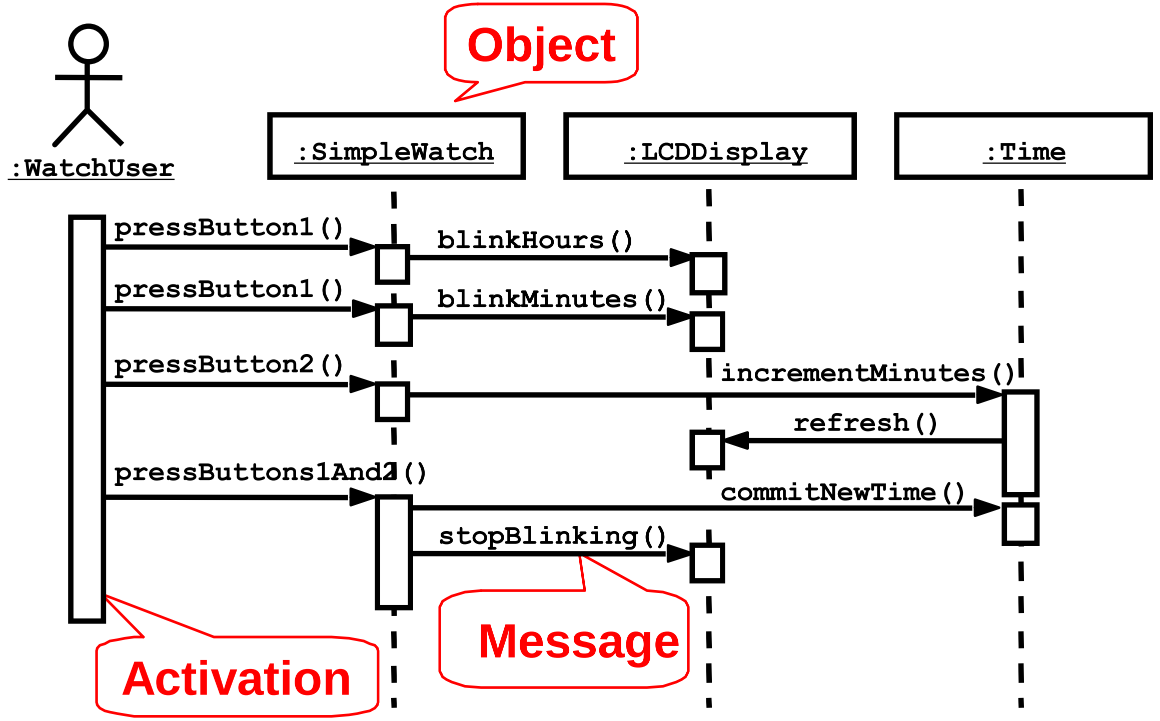
- The source of an arrow indicates the activation which sent the message
- An activation is as long as all nested activations

Sequence Diagram Observations

- UML sequence diagram represent **behavior** in terms of **interactions**
  - ***Run-time view***
- They complement class diagrams, which represent **structure**
  - ***Compile-time view***
- Very useful for **finding participating objects**
- Time-consuming to build but worth the investment where interactions are complex

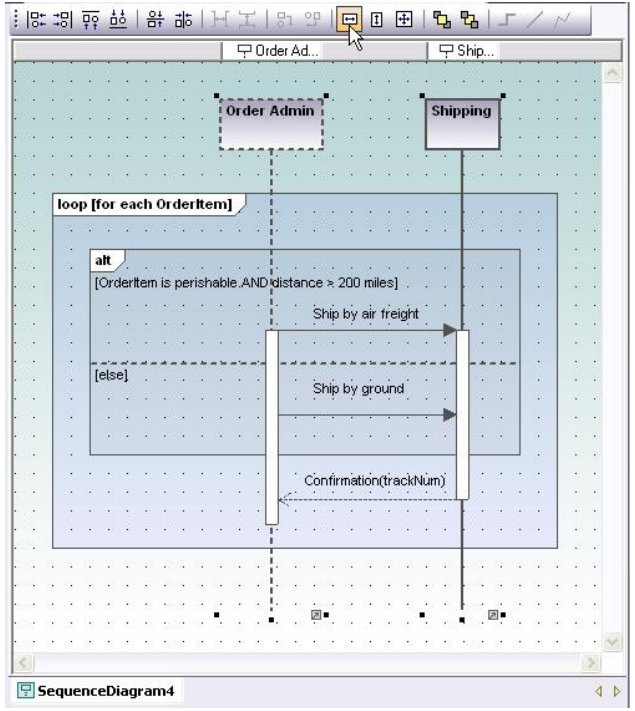
LULEÅ UNIVERSITY OF TECHNOLOGY

Sequence Diagram



Sequence Diagrams – Interaction Frames

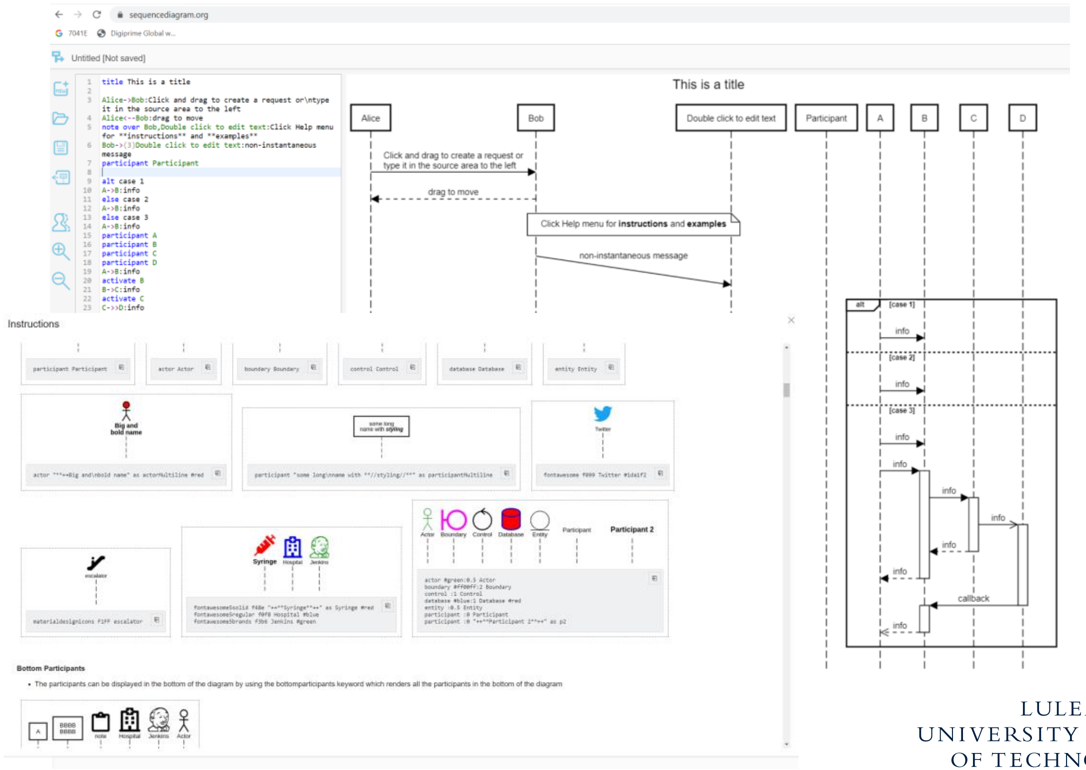
- alt (conditional)
- opt (optional)
- par (run in parallel)
- loop (iteration)
- region (one thread)
- neg (invalid)
- ref (reference: defined in another diagram)



LULEÅ UNIVERSITY OF TECHNOLOGY

Sequence Diagrams – Tools

- **Sequencediagram.org**
- Lucidchart
  - Flow chart diagram
  - Sequence diagram
  - Class diagram
  - **State chart (premium)**
  - *Gantt chart*
  - *Roadmap*
- Visual Paradigm online
  - Most (all?) UML diagrams
  - UML State chart (machine) Diagram
- *and more…*



```
title This is a title

Alice->Bob:Click and drag to create a request or\ntype it in the source area to the left
Alice<--Bob:drag to move
note over Bob,Double click to edit text:Click Help menu for **instructions** and **examples**
Bob->(3)Double click to edit text:non-instantaneous message
participant Participant

alt case 1
A->B:info
else case 2
A->B:info
else case 3
A->B:info
participant A
participant B
participant C
participant D
A->B:info
activate B
B->C:info
activate C
C->>D:info
```

```
participant Participant
```

```
actor Actor
```

```
boundary Boundary
```

```
control Control
```

```
database Database
```

```
entity Entity
```

```
actor "**++Big and\nbold name" as actorMultiline #red
```

```
participant "some long\nname with **//styling//**" as participantMultiline
```

```
fontawesome f099 Twitter #1da1f2
```

```
materialdesignicons F1FF escalator
```

```
fontawesome5solid f48e "++**Syringe**++" as Syringe #red
fontawesome5regular f0f8 Hospital #blue
fontawesome5brands f3b6 Jenkins #green
```

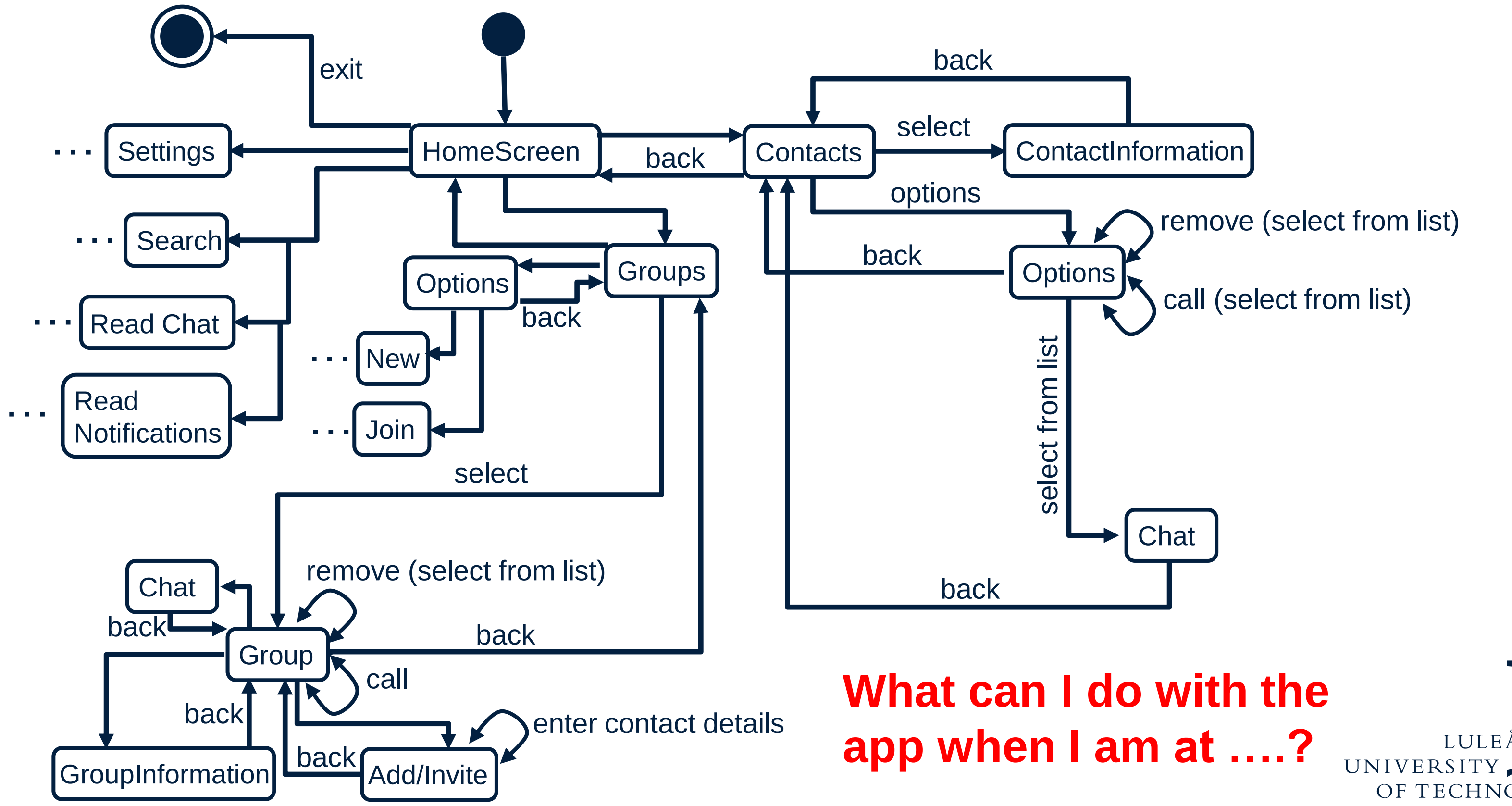
```
actor #green:0.5 Actor
boundary #ff00ff:2 Boundary
control :1 Control
database #blue:1 Database #red
entity :0.5 Entity
participant :0 Participant
participant :0 "++**Participant 2**++" as p2
```

Bottom Participants

- The participants can be displayed in the bottom of the diagram by using the bottomparticipants keyword which renders all the participants in the bottom of the diagram

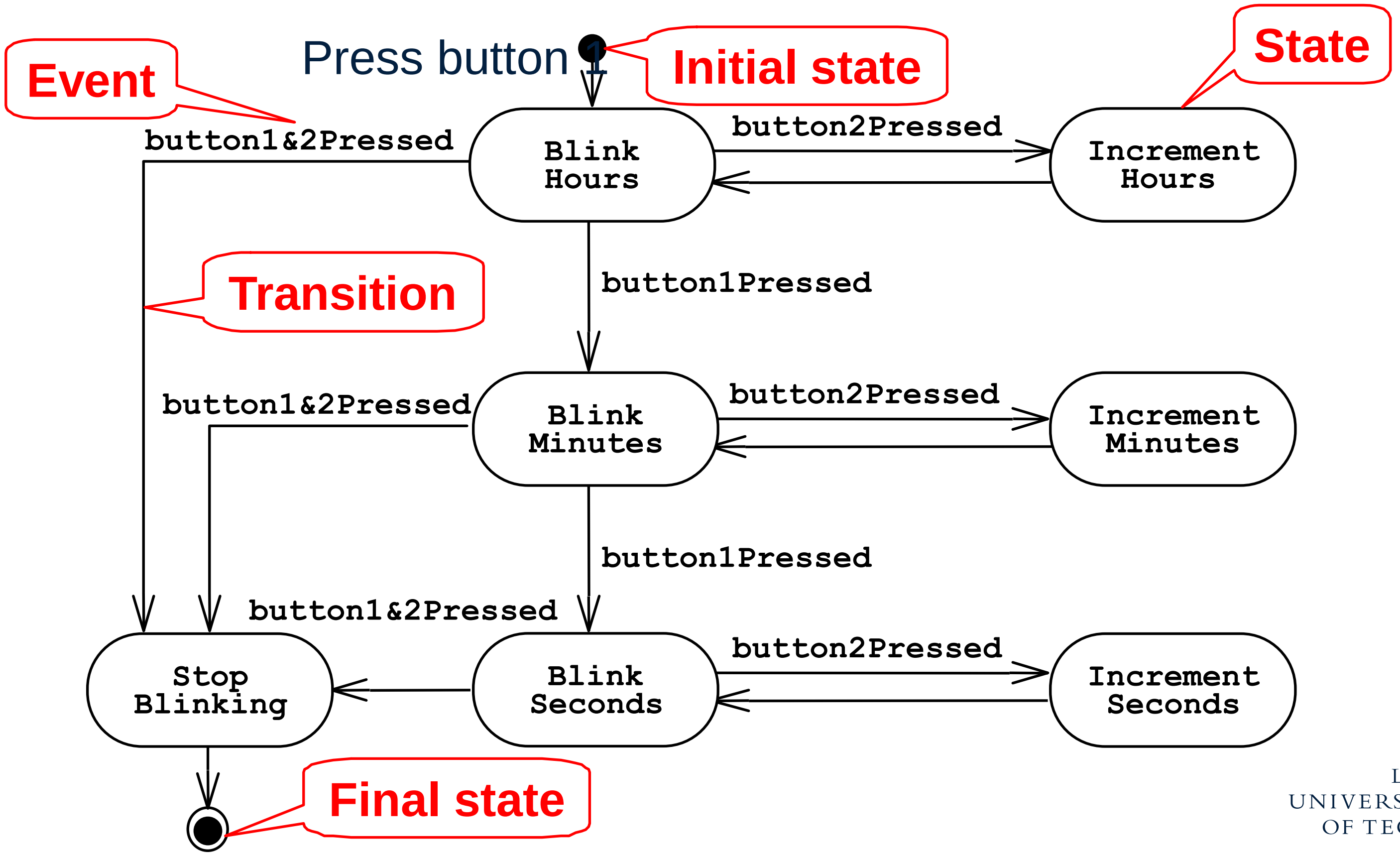
LULEÅ UNIVERSITY OF TECHNOLOGY

State chart (machine) Diagrams



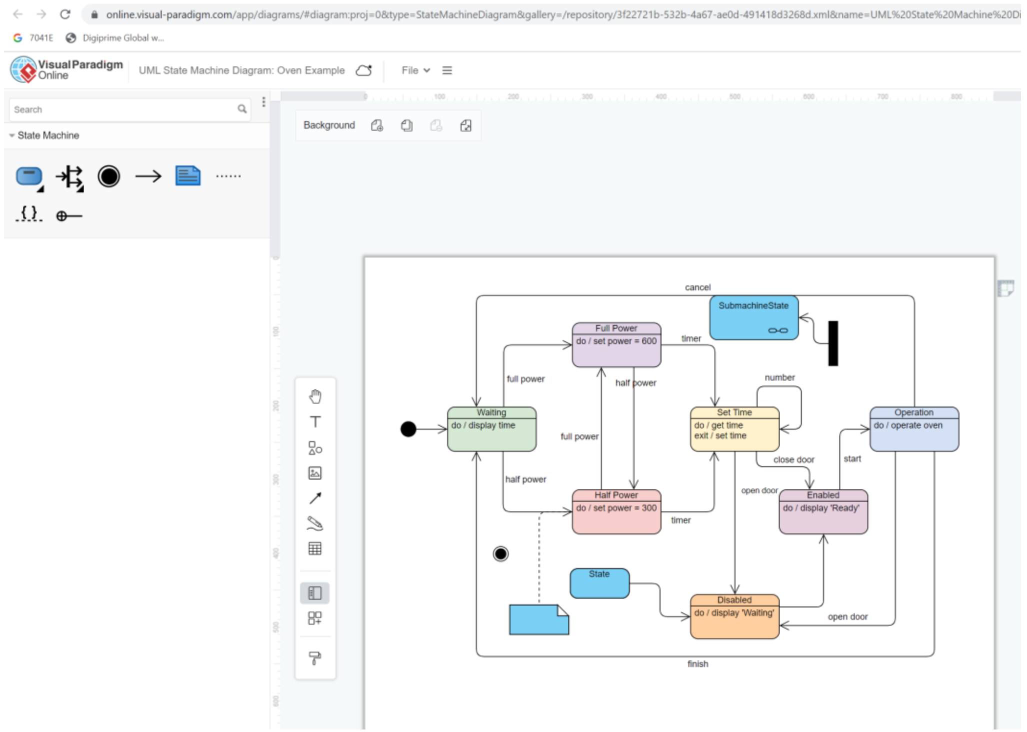
What can I do with the app when I am at ….?

State chart (machine) Diagrams



State chart (machine) Diagrams – Tools

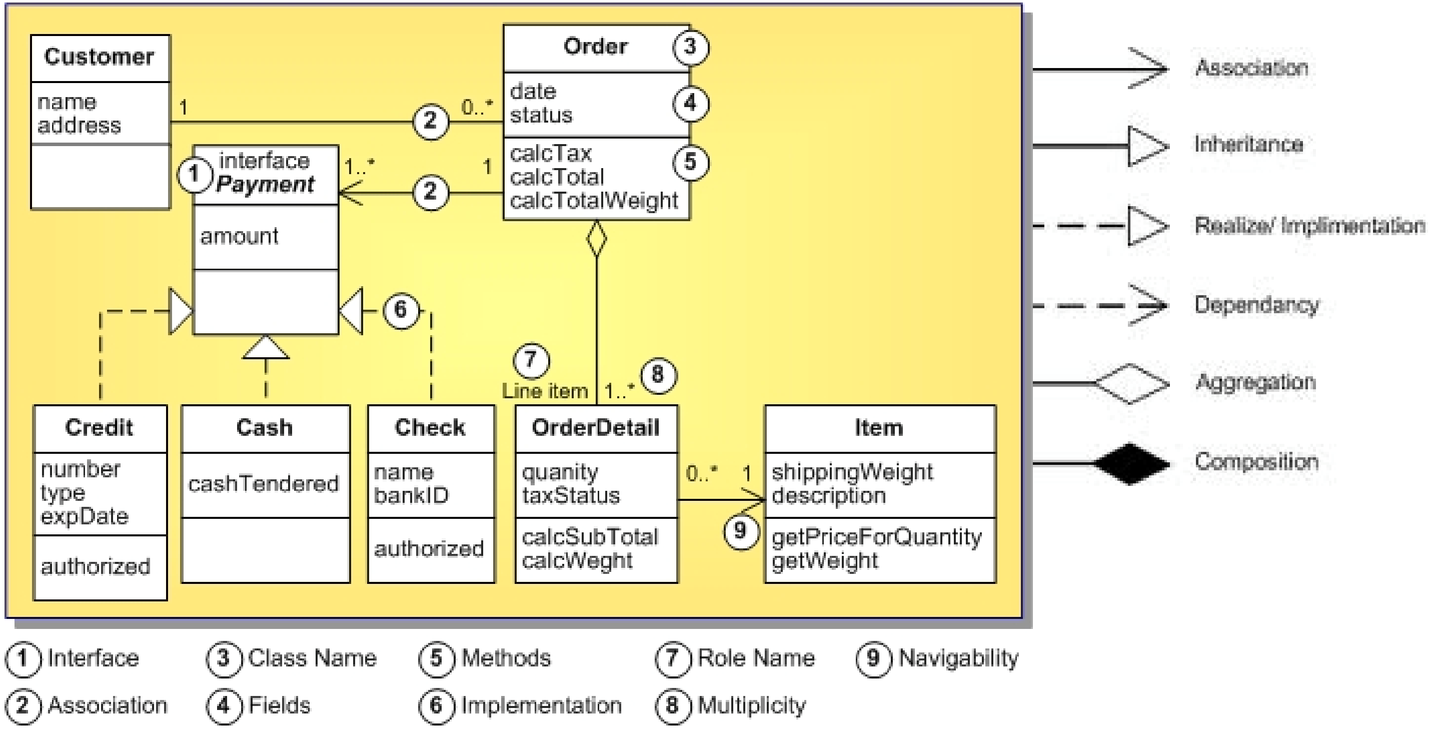
Visual Paradigm:



Simple diagram goes far, do not overdo it with advanced tools…

LULEÅ UNIVERSITY OF TECHNOLOGY

Class diagram

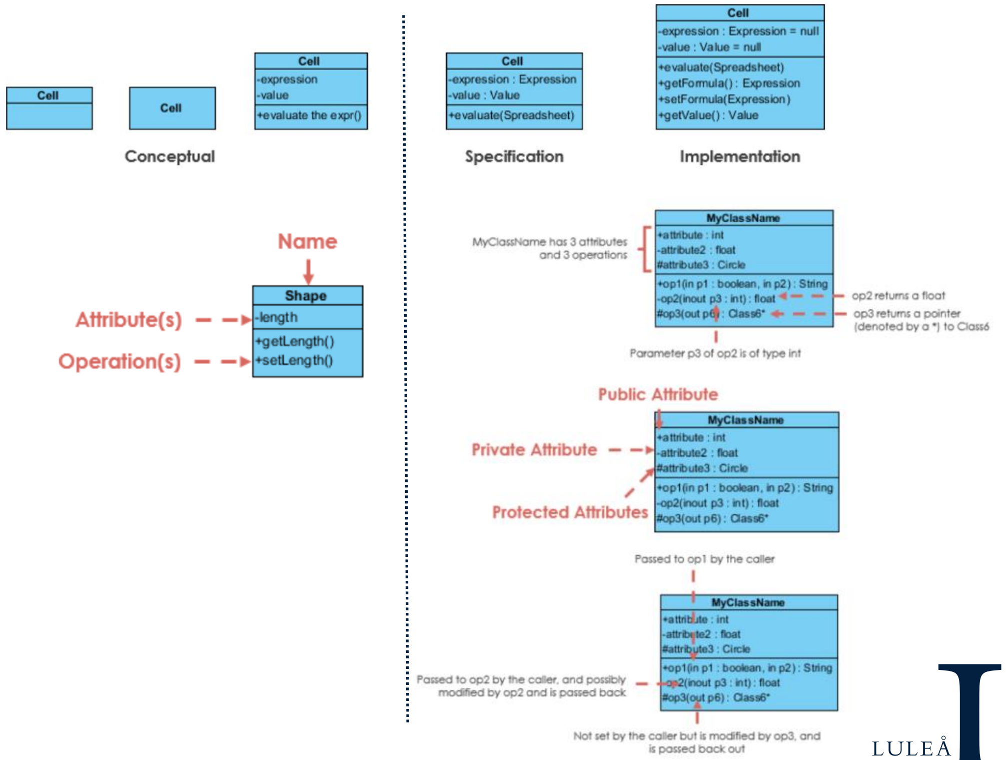


LULEÅ UNIVERSITY OF TECHNOLOGY

Class diagram – usage and example

Perspectives (abstraction)

- Conceptual
  - Represents the concepts in the domain
- Specification
  - Focus is on the interfaces of Abstract Data Type (ADTs) in the software
- Implementation
  - Describes how classes will implement their interfaces



* Visual Paradigm, UML Class Diagram Tutorial. URL (accessed 2022-11-22): https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/

LULEÅ UNIVERSITY OF TECHNOLOGY

Summary

- UML provides a wide variety of notations for representing many aspects of software development
  - Powerful, but complex language (keep it simple!)
  - Can be misused to generate unreadable models (automatic generators)
  - Can be misunderstood when using too many exotic features (KISS!)

- We concentrate only on a few notations:
  - **Functional model**: **use case diagram**
  - **Object model**: **class diagram**
  - **Dynamic model**: **sequence, state chart, and activity diagrams**

- UML is not a methodology, but some textbooks describe a methodology that <u>uses</u> UML

LULEÅ
UNIVERSITY
OF TECHNOLOGY

BREAK

IT'S TIME WE TAKE A BREAK.
YOU'RE RIGHT. LET'S GO WORK FOR A LITTLE WHILE.
POITRAS

LULEÅ UNIVERSITY OF TECHNOLOGY

LULEÅ
UNIVERSITY
OF TECHNOLOGY