TEACHING

D0020E PROJECT IN COMPUTER SCIENCE 2022/2023 LECTURE 6.1: MODELING

Ulf Bodin

LTU 22.11.2022

ONE OUT OF TEN RESEARCH AND DEVELOPMENT PROJECTS WILL SUCCEED.
I RECOMMEND CANCELLING THE OTHER NINE.
I WONDER WHERE HE GETS ALL THESE CRAZY IDEAS.
www.dilbert.com scottadams@aol.com
© 2000 United Feature Syndicate, Inc.

LULEÅ UNIVERSITY OF TECHNOLOGY

This lecture

- The beginning of the development process...
  - The Uniform Modeling Language (UML)
  - The System Expert and The User Group
  - Actors
  - Use Cases
  - Use Case Diagrams
  - Activity Diagrams
  - Storyboards
  - Roles and Responsiblities

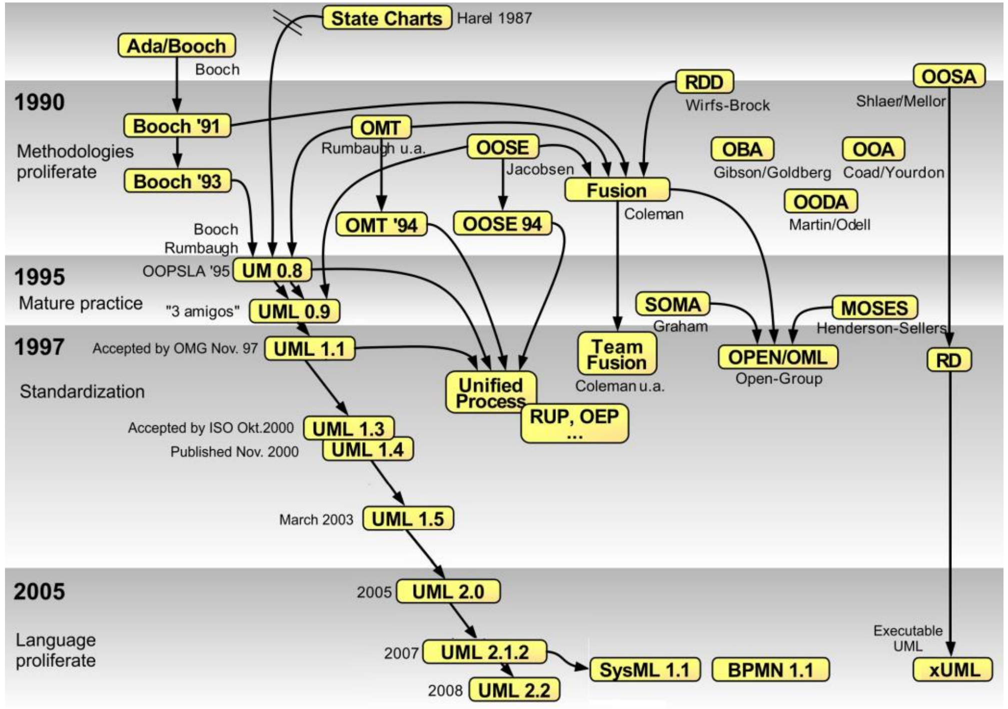
LULEÅ UNIVERSITY OF TECHNOLOGY

Modeling using UML

DeFacto standard for modeling object-oriented systems.

Resulted from the convergence of notations from three leading object-oriented methods:

- OMT (James Rumbaugh), **development processes**
- OOSE (Ivar Jacobson), objectory or **use-case driven design**
- Booch (Grady Booch), **notation**

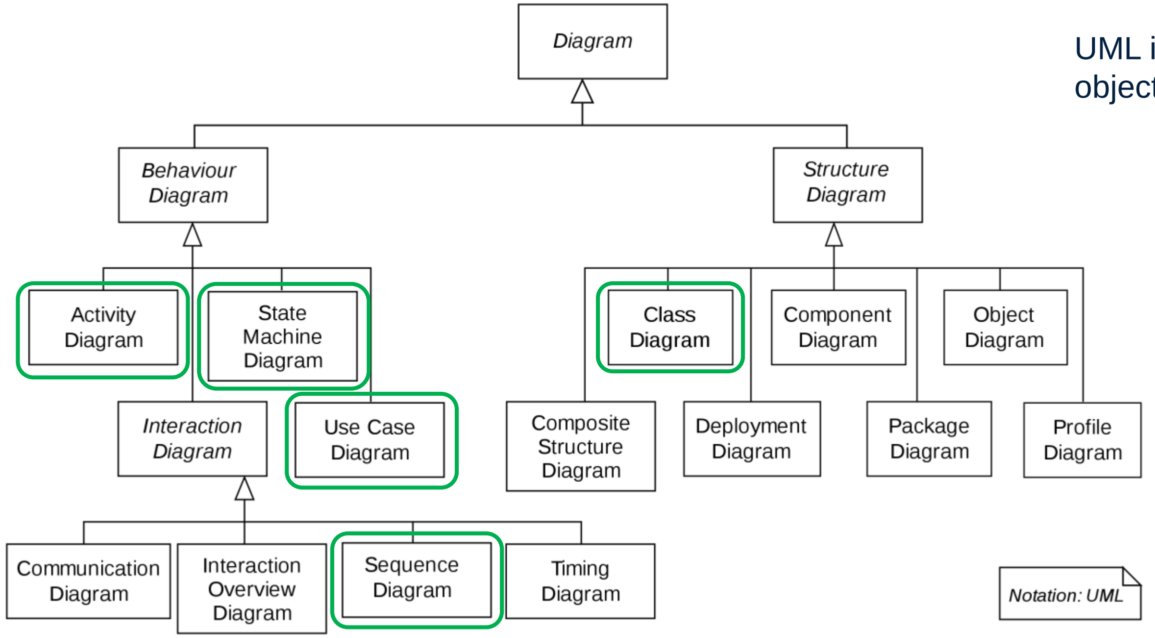


*

* Wikipedia, Unified Modeling Language. URL (accessed 2022-11-21): https://en.wikipedia.org/wiki/Unified_Modeling_Language

LULEÅ UNIVERSITY OF TECHNOLOGY

UML Diagrams

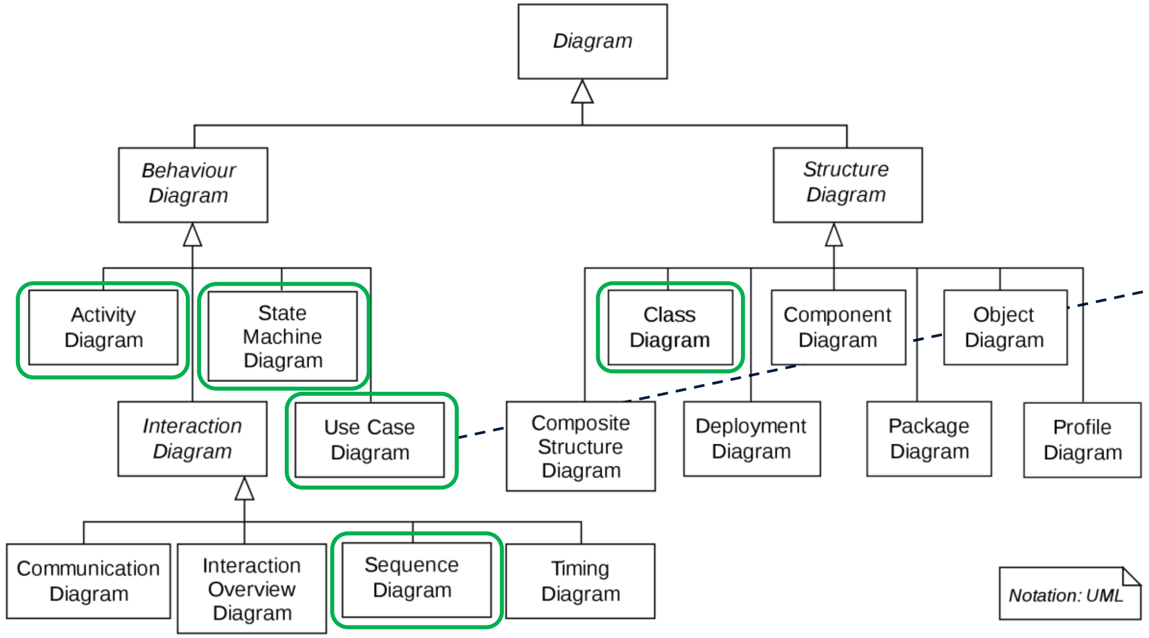


UML is designed to support most existing object-oriented development processes

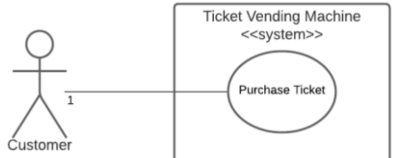
- ***Structure diagrams***
  - Capture static aspects
    - What must be present
  - Document the software **architecture**
- ***Behavior diagrams***
  - Capture dynamic aspects
    - What must happen
  - Describe the **functionality** of software systems
- ***Interaction diagrams***
  - The **flow** of control/data
    - E.g., the sequence diagram message communication between objects

* Wikipedia, Unified Modeling Language. URL (accessed 2022-11-21): https://en.wikipedia.org/wiki/Unified_Modeling_Language

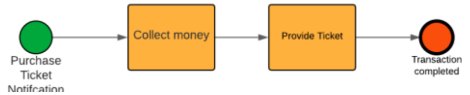
UML vs. BPMN



UML is designed to support most existing object-oriented development processes

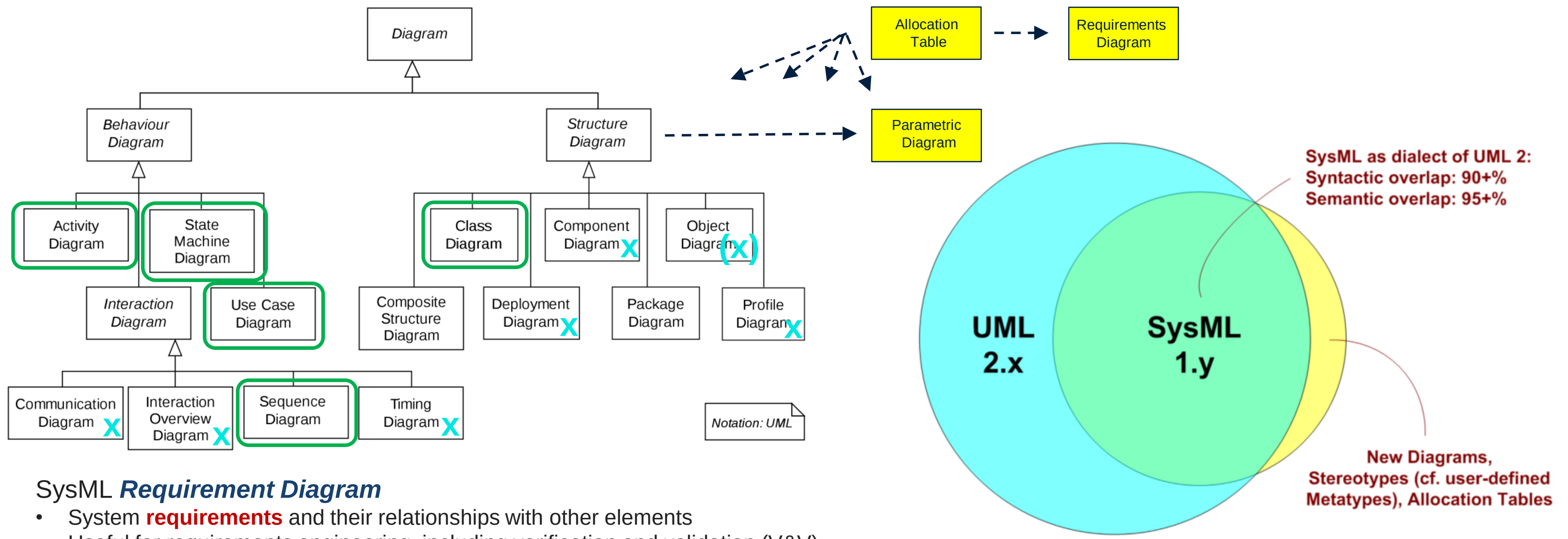


Business Process Model and Notation (BPMN) is process oriented



* Wikipedia, Unified Modeling Language. URL (accessed 2022-11-21): https://en.wikipedia.org/wiki/Unified_Modeling_Language

UML vs. SysML



SysML ***Requirement Diagram***

- System **requirements** and their relationships with other elements
- Useful for requirements engineering, including verification and validation (V&V)

SysML ***Parametric Diagram***

- Parametric **constraints** between structural elements
- Useful for performance and quantitative analysis

SysML ***Mapping Table***; not a diagram

- Assignment **relationships** between elements.
- Useful for facilitating automated V&V and gap analysis

* Wikipedia, Unified Modeling Language. URL (accessed 2022-11-21): https://en.wikipedia.org/wiki/Unified_Modeling_Language
** SysML.org, SysML FAQ: What is the relation between SysML and UML? URL (accessed 2022-11-21): https://sysml.org/sysml-faq/what-is-relation-between-sysml-and-uml.html

LULEÅ UNIVERSITY OF TECHNOLOGY

Modeling using UML

Modeling is a central part of all the activities that lead up to the deployment of good software.

We use models to:

- Understand the system as a whole and in parts
- Visualize and control the system’s architecture
- Communicate desired system structures and behavior
- To identify cases of abstraction and reuse
- Manage risks, and build test strategy

You can model 80% of most problems by using ~20% UML!

LULEÅ UNIVERSITY OF TECHNOLOGY

Modeling using UML

- Models abstracts the complexity of systems so they become understandable.

- The models are often expressed as:
  - Text
  - Diagrams
  - Structured Data

The four basic principles of modeling:

1. The choice of what models to create has a profound influence on how a problem is attacked and how a solution is shaped.
2. Every model may be expressed at different levels of precision.
3. The best models are connected to reality (models an abstracted reality).
4. No single model is sufficient, as every nontrivial system is best approached through a small set of nearly independent models.

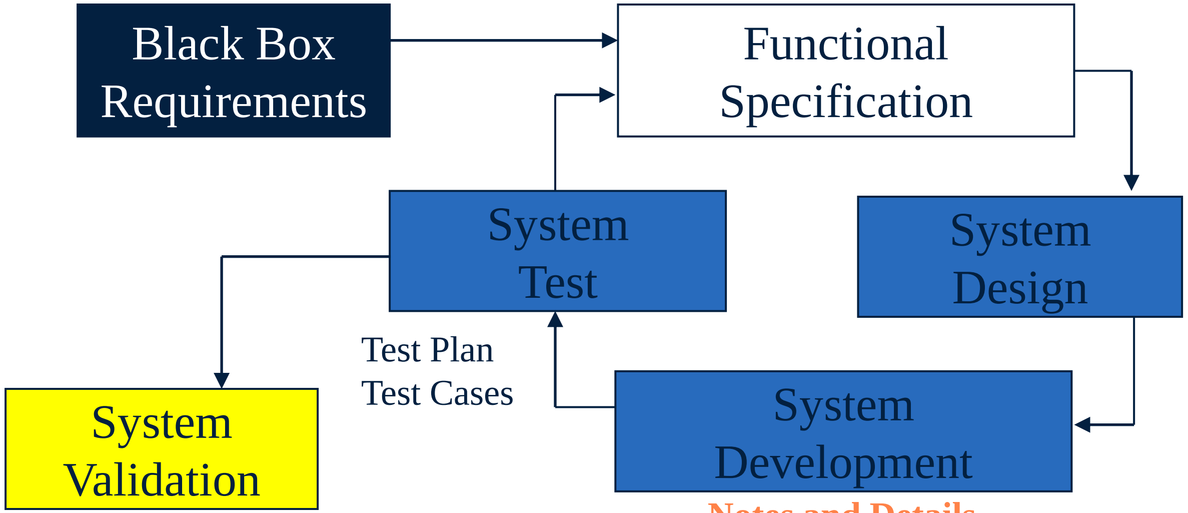
Modeling using UML

Diagrams are the main artefacts when using 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.
- **Sequence diagrams**
  - Dynamic behavior between actors and system objects.
- **State-chart (machine) diagrams**
  - Dynamic behavior of an individual object.
- **Class diagrams**
  - Static structure of the system: Objects, Attributes, and Associations.
- etc...

LULEÅ UNIVERSITY OF TECHNOLOGY

Artifacts



Actors
Use Cases
Use Case Diagrams
Storyboards
Requirements Specification

Test Plan
Test Cases

Diagrams:
- Class
- Sequence
- Statechart
- Activity
Design Patterns
System Design

Notes and Details
Signatures
Design Patterns

LULEÅ UNIVERSITY OF TECHNOLOGY

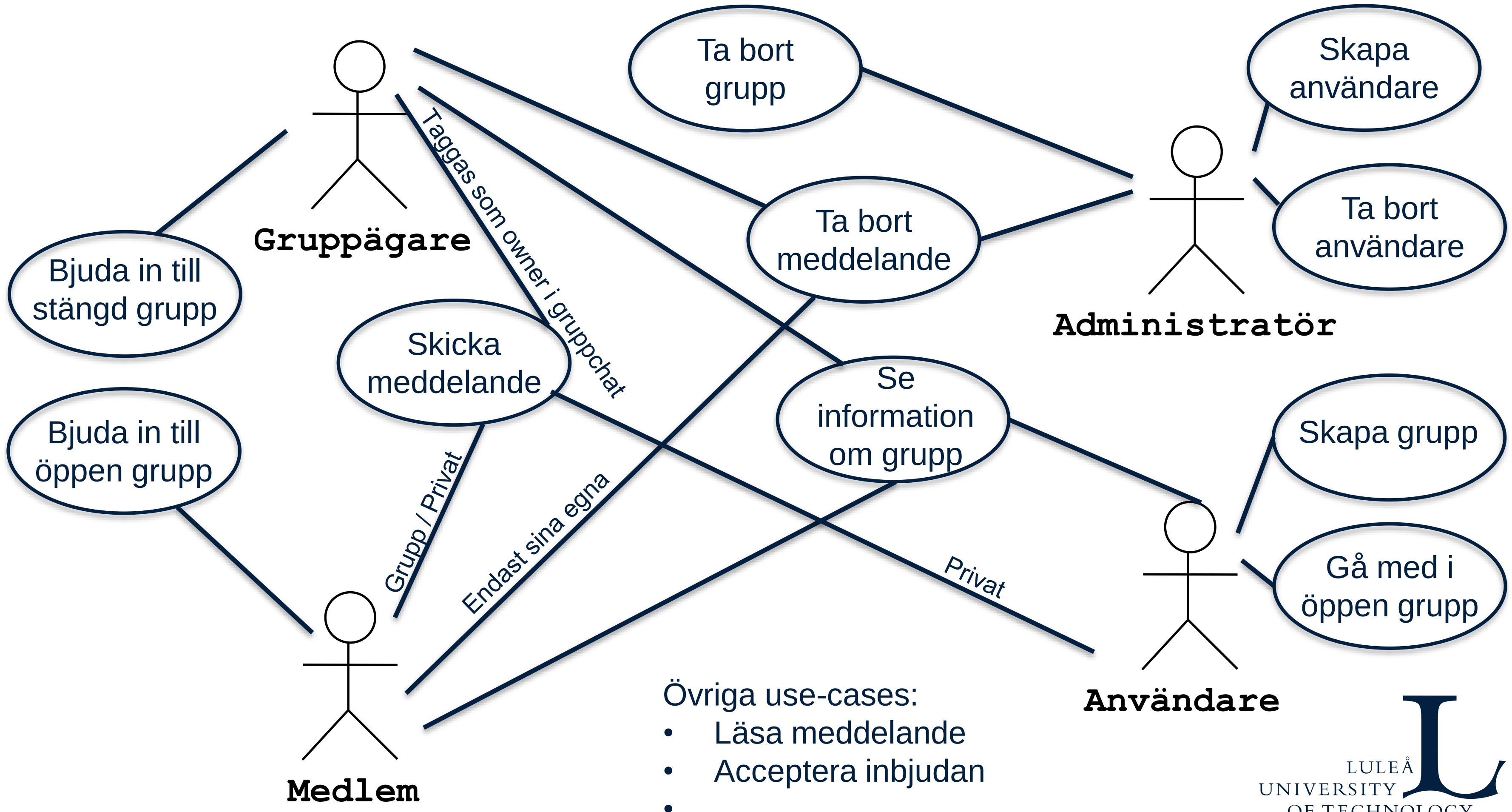
System Experts / Product Owner

- You cannot do a good design if the main requirements are unknown! Feedback from the users are **vital**.
- A **system expert / system architect / product owner** can give you invaluable help in the design of your system, especially when it comes to **interaction** and **usage**.
  - The system expert can be, and is likely to be, a user.
- Use the system expert especially to identify the use cases of the system.

Example

- How do you express the functions and the requirements?

- 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?



Övriga use-cases:

- Läsa meddelande
- Acceptera inbjudan
- …

LULEÅ UNIVERSITY OF TECHNOLOGY

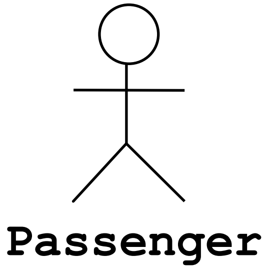
Användarfallsbeskrivning

- 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

To do

- Användarfallsbeskrivning
  - Bjud in till grupp
  - Aktörer: Gruppägare, Användare
- Aktivitetsdiagram
  - Bjud in till grupp
  - Swimlanes: acceptera inbjudan av Användare
- Sekvensdiagram
  - Bjud in till grupp
  - Authenticate, DB contact info, Listened Användare, etc.
- Tillståndsdiagram
  - Group communication app

Actors



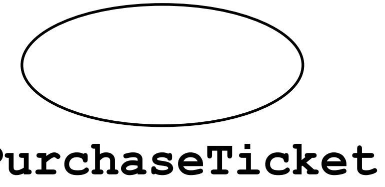
- An actor models an external entity which communicates with the system:
  - User
  - *External system*
  - *Physical environment*

- An actor has a unique name and an optional description.

- Examples:
  - Passenger: A person in the train
  - *GPS satellite: Provides the system with GPS coordinates*

LULEÅ UNIVERSITY OF TECHNOLOGY

Use Case



A use case represents a class of functionality provided by the system as an event flow.

A use case consists of:

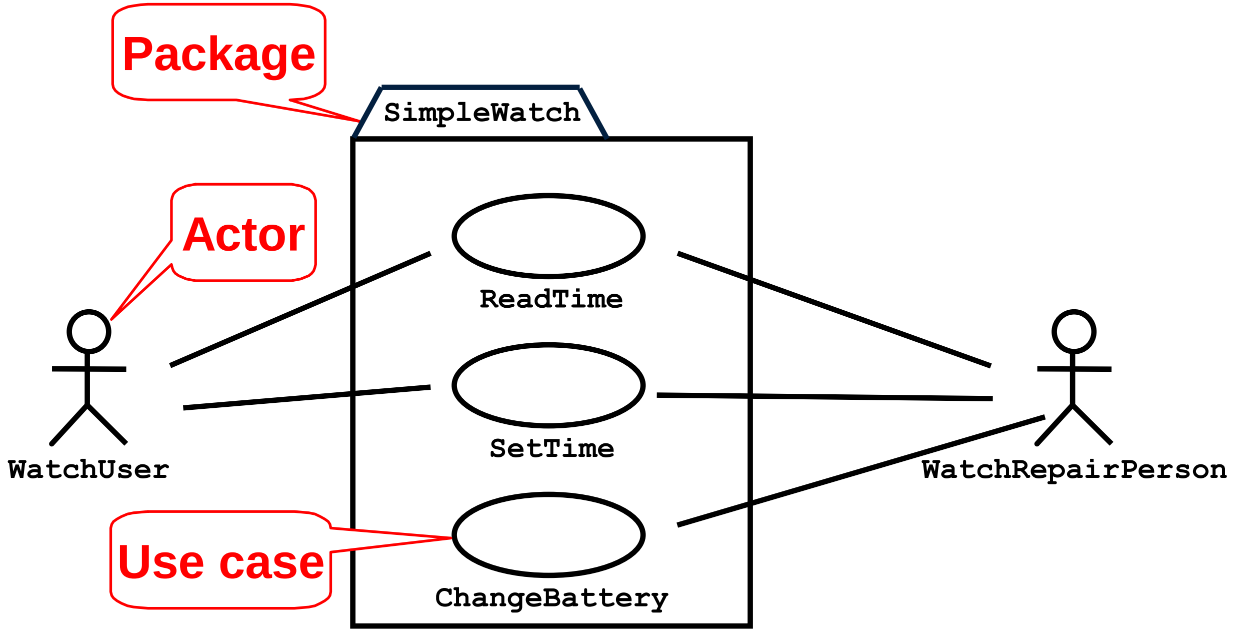
- Unique name
- Participating actors
- **Entry conditions**
- Flow of events
- **Exit conditions**
- Special requirements

Use Cases

- A use case is basically a functional view of the system
  - The print method
  - Finding a webpage
- The most commonly used use cases are the most important to get right!
- Do not try to capture all the details right at the start – allow for iteration and discussion!
- Use cases should also affect how the system is tested during development!

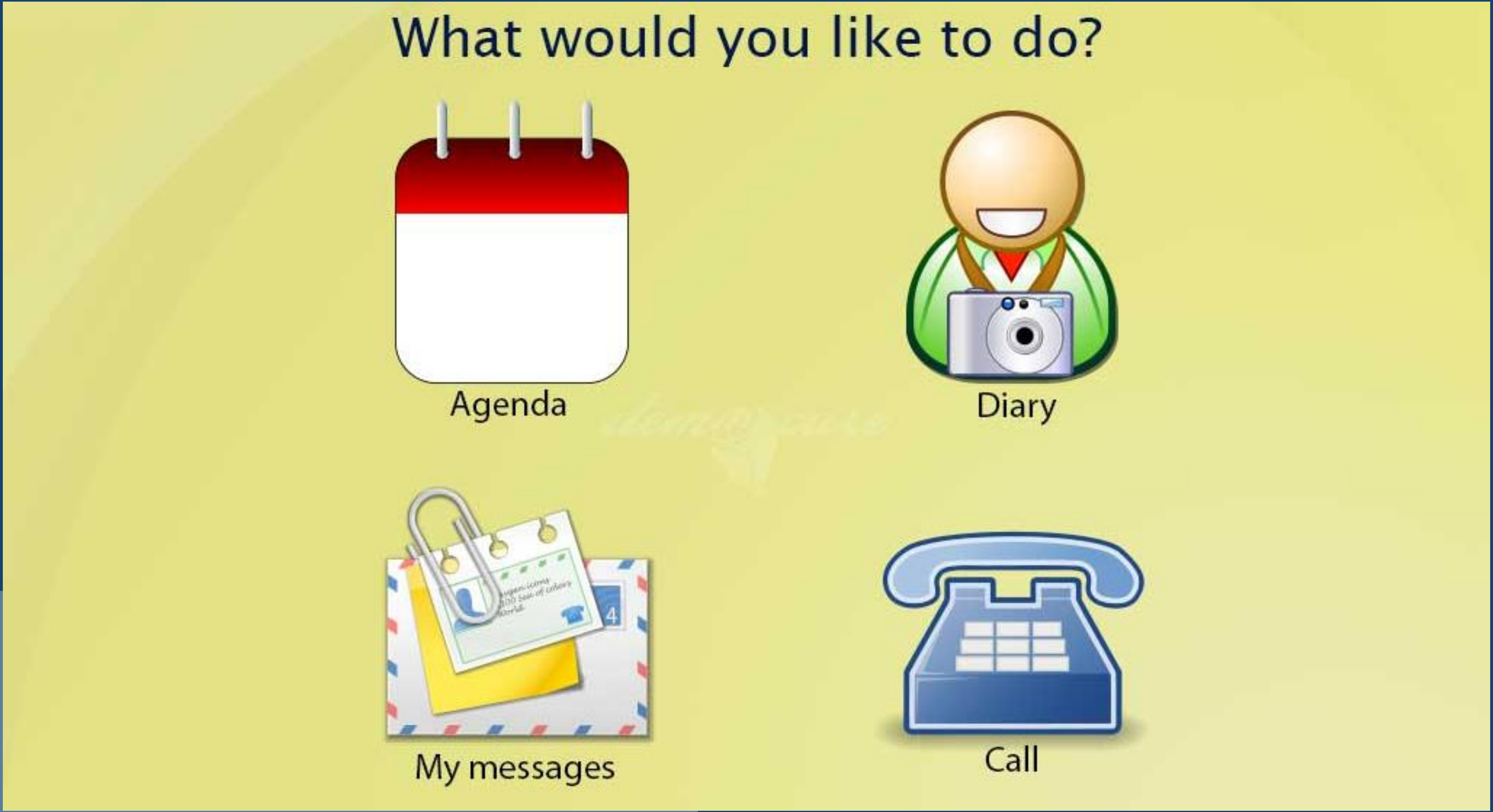
LULEÅ UNIVERSITY OF TECHNOLOGY

Use Case Diagrams



Use case diagrams represent the functionality of the system from user’s point of view.

Dem@Care



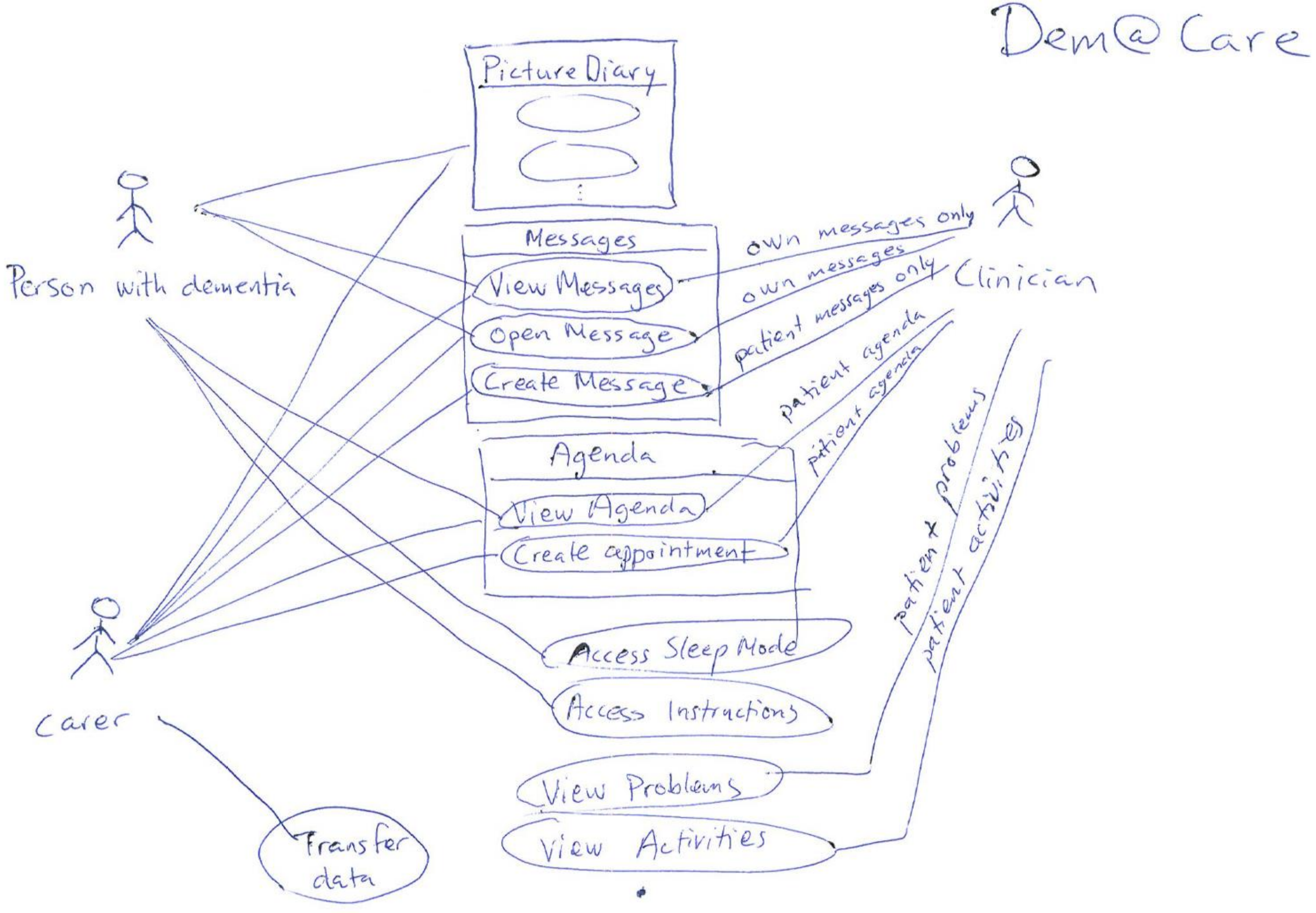
What would you like to do?

Agenda

Diary

My messages

Call


Dem@Care
Picture Diary
Messages
View Messages
Open Message
Create Message
Agenda
View Agenda
Create appointment
Access Sleep Mode
Access Instructions
View Problems
View Activities
Transfer data
Person with dementia
Carer
Clinician
own messages only
own messages
patient messages only
patient agenda
patient agenda
patient problems
patient activities

Use Case Example 1

Name: `Change Battery`

Participating actors:
`WatchRepairPerson`

Entry condition:

- `WatchRepairPerson` has a new battery.

Exit condition:

- `WatchRepairPerson` has changed to a new battery.

Event flow:

1. `WatchRepairPerson` opens the clock and replaces the old battery with a new battery.

LULEÅ UNIVERSITY OF TECHNOLOGY

Use Case Example 2

Name: `Purchase ticket`

Participating actor: `Passenger`

Entry condition:

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

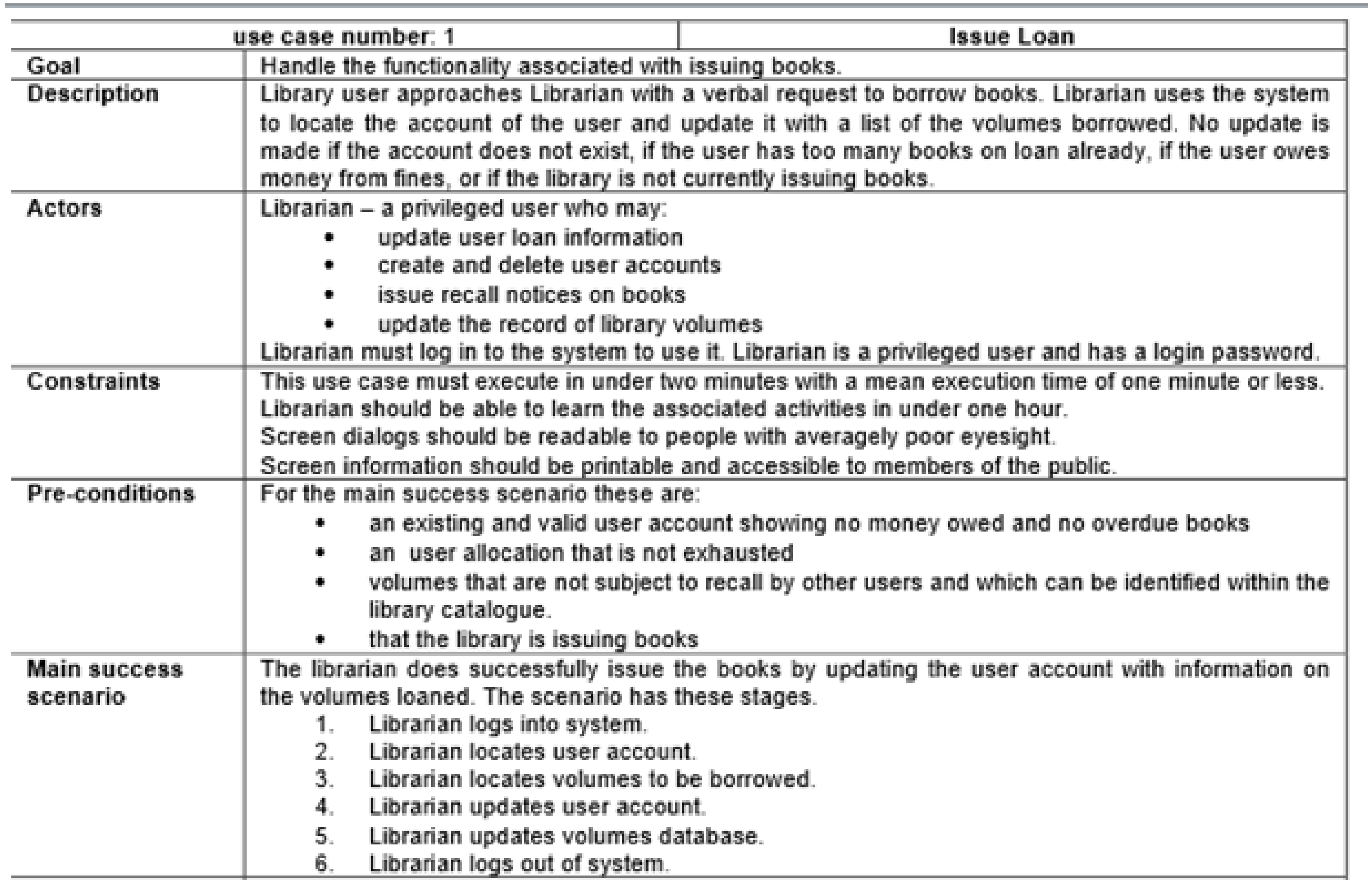
Exit condition:

- `Passenger` has ticket.

Event flow:

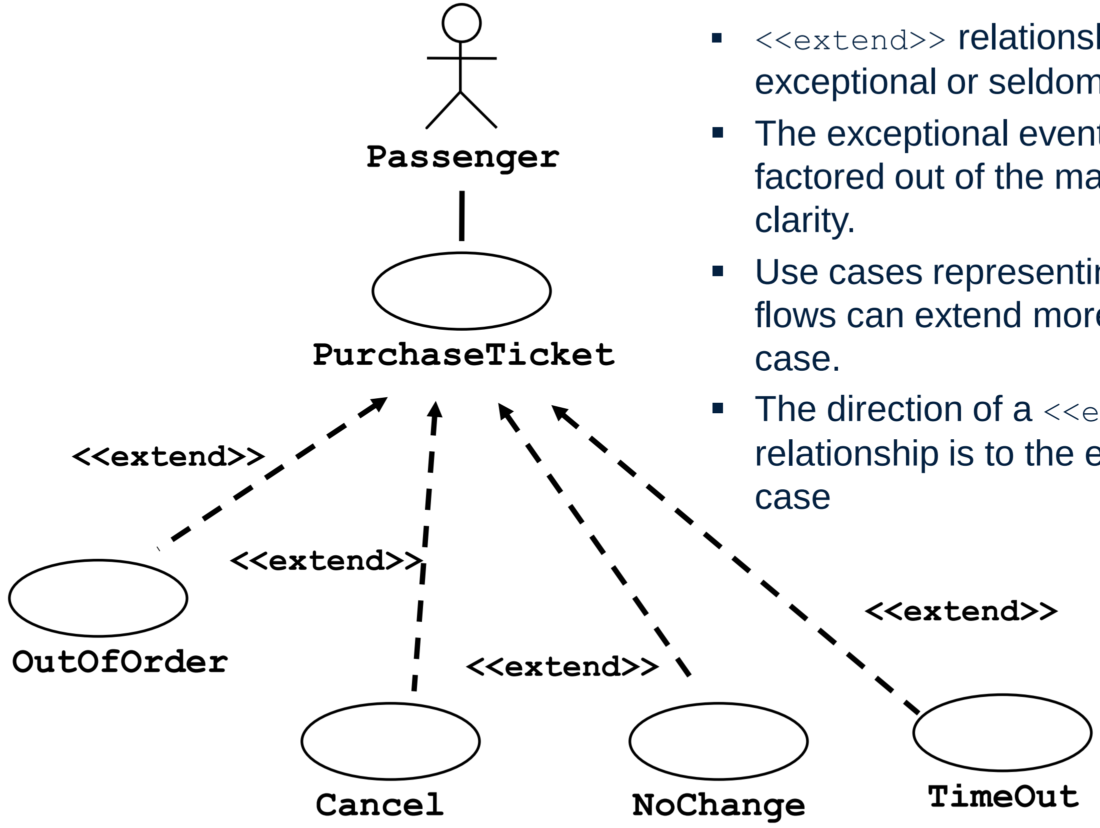
1. `Passenger` selects the number of zones to be traveled.
2. Distributor displays the amount due.
3. `Passenger` inserts money, of at least the amount due.
4. Distributor returns change.
5. Distributor issues ticket.

LULEÅ UNIVERSITY OF TECHNOLOGY



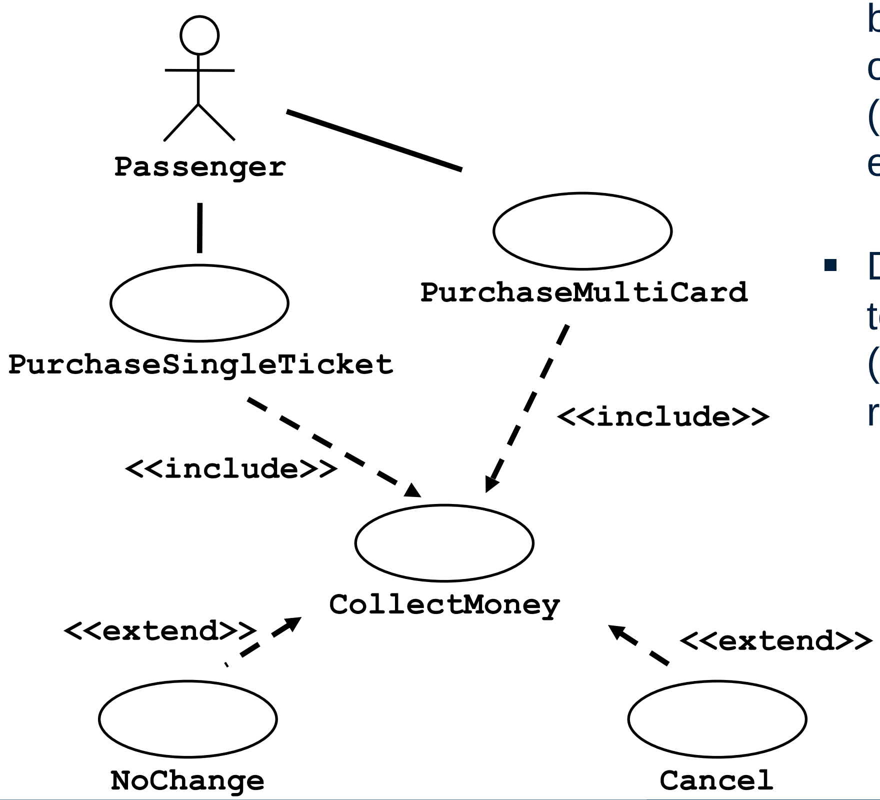
use case number: 1	Issue Loan
Goal	Handle the functionality associated with issuing books.
Description	Library user approaches Librarian with a verbal request to borrow books. Librarian uses the system to locate the account of the user and update it with a list of the volumes borrowed. No update is made if the account does not exist, if the user has too many books on loan already, if the user owes money from fines, or if the library is not currently issuing books.
Actors	Librarian – a privileged user who may: • update user loan information • create and delete user accounts • issue recall notices on books • update the record of library volumes Librarian must log in to the system to use it. Librarian is a privileged user and has a login password.
Constraints	This use case must execute in under two minutes with a mean execution time of one minute or less. Librarian should be able to learn the associated activities in under one hour. Screen dialogs should be readable to people with averagely poor eyesight. Screen information should be printable and accessible to members of the public.
Pre-conditions	For the main success scenario these are: • an existing and valid user account showing no money owed and no overdue books • an user allocation that is not exhausted • volumes that are not subject to recall by other users and which can be identified within the library catalogue. • that the library is issuing books
Main success scenario	The librarian does successfully issue the books by updating the user account with information on the volumes loaned. The scenario has these stages. 1. Librarian logs into system. 2. Librarian locates user account. 3. Librarian locates volumes to be borrowed. 4. Librarian updates user account. 5. Librarian updates volumes database. 6. Librarian logs out of system.

The <<extend>> Relationship



- <<extend>> relationships represent exceptional or seldom invoked cases.
- The exceptional event flows are factored out of the main event flow for clarity.
- Use cases representing exceptional flows can extend more than one use case.
- The direction of a <<extend>> relationship is to the extended use case

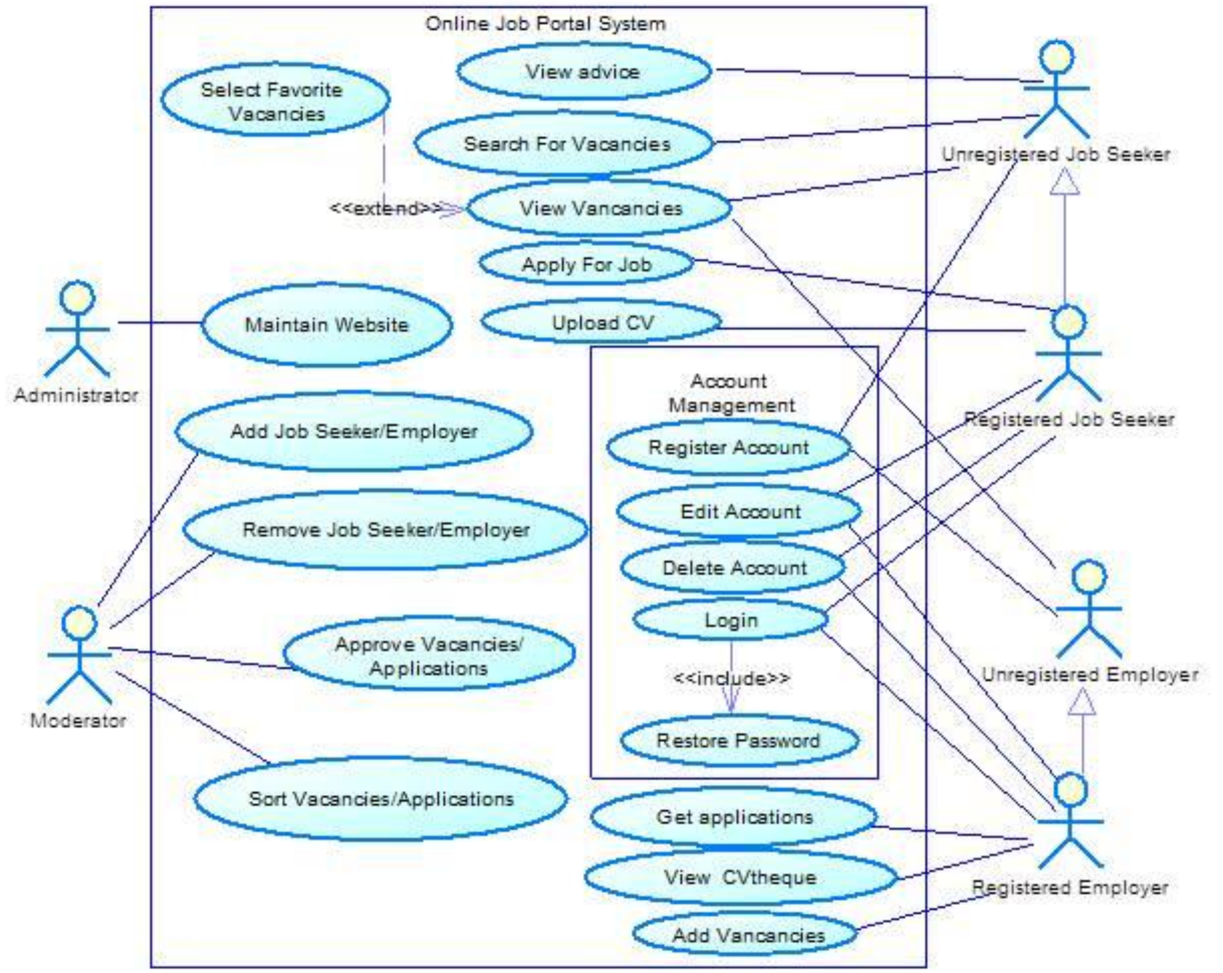
The <<include>> Relationship



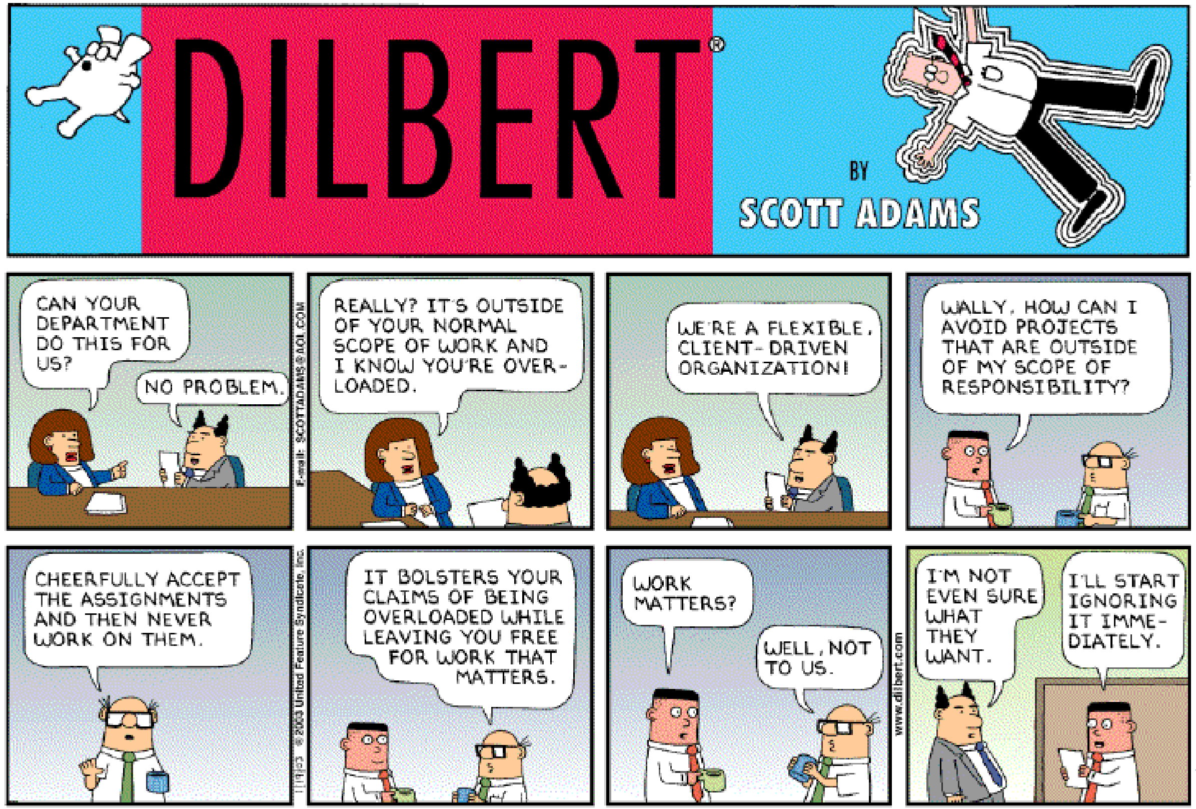
- `<<include>>` represents behavior that is factored out of the use case for reuse (not because it is an exception)

- Direction of `<<include>>` is to the using use case (unlike `<<extend>>` relationships).

LULEÅ UNIVERSITY OF TECHNOLOGY


Online Job Portal System
Select Favorite Vacancies
View advice
Search For Vacancies
<<extend>>
View Vancancies
Apply For Job
Upload CV
Maintain Website
Administrator
Account Management
Register Account
Edit Account
Delete Account
Login
<<include>>
Restore Password
Add Job Seeker/Employer
Remove Job Seeker/Employer
Approve Vacancies/ Applications
Moderator
Sort Vacancies/Applications
Get applications
View CVtheque
Add Vancancies
Unregistered Job Seeker
Registered Job Seeker
Unregistered Employer
Registered Employer

Break


DILBERT®
BY SCOTT ADAMS
CAN YOUR DEPARTMENT DO THIS FOR US?
NO PROBLEM.
E-mail: SCOTTADAMS@AOL.COM
REALLY? IT'S OUTSIDE OF YOUR NORMAL SCOPE OF WORK AND I KNOW YOU'RE OVER-LOADED.
WE'RE A FLEXIBLE, CLIENT-DRIVEN ORGANIZATION!
WALLY, HOW CAN I AVOID PROJECTS THAT ARE OUTSIDE OF MY SCOPE OF RESPONSIBILITY?
CHEERFULLY ACCEPT THE ASSIGNMENTS AND THEN NEVER WORK ON THEM.
© 2003 United Feature Syndicate, Inc.
IT BOLSTERS YOUR CLAIMS OF BEING OVERLOADED WHILE LEAVING YOU FREE FOR WORK THAT MATTERS.
WORK MATTERS?
WELL, NOT TO US.
www.dilbert.com
I'M NOT EVEN SURE WHAT THEY WANT.
I'LL START IGNORING IT IMME-DIATELY.

LULEÅ UNIVERSITY OF TECHNOLOGY

LULEÅ
UNIVERSITY
OF TECHNOLOGY