Software Requirement Concepts

greenwich.edu.vn

UNIVERSITY of GREENWICH
Alliance with FPT Education

Agenda

- Requirement Concepts
- Requirement Modeling
- Modeling Tools:
  - OOAD: Use Case, Activity Diagram, State Machine Diagram
  - SSADM: Data Flow Diagram, Entity Relationship Diagram

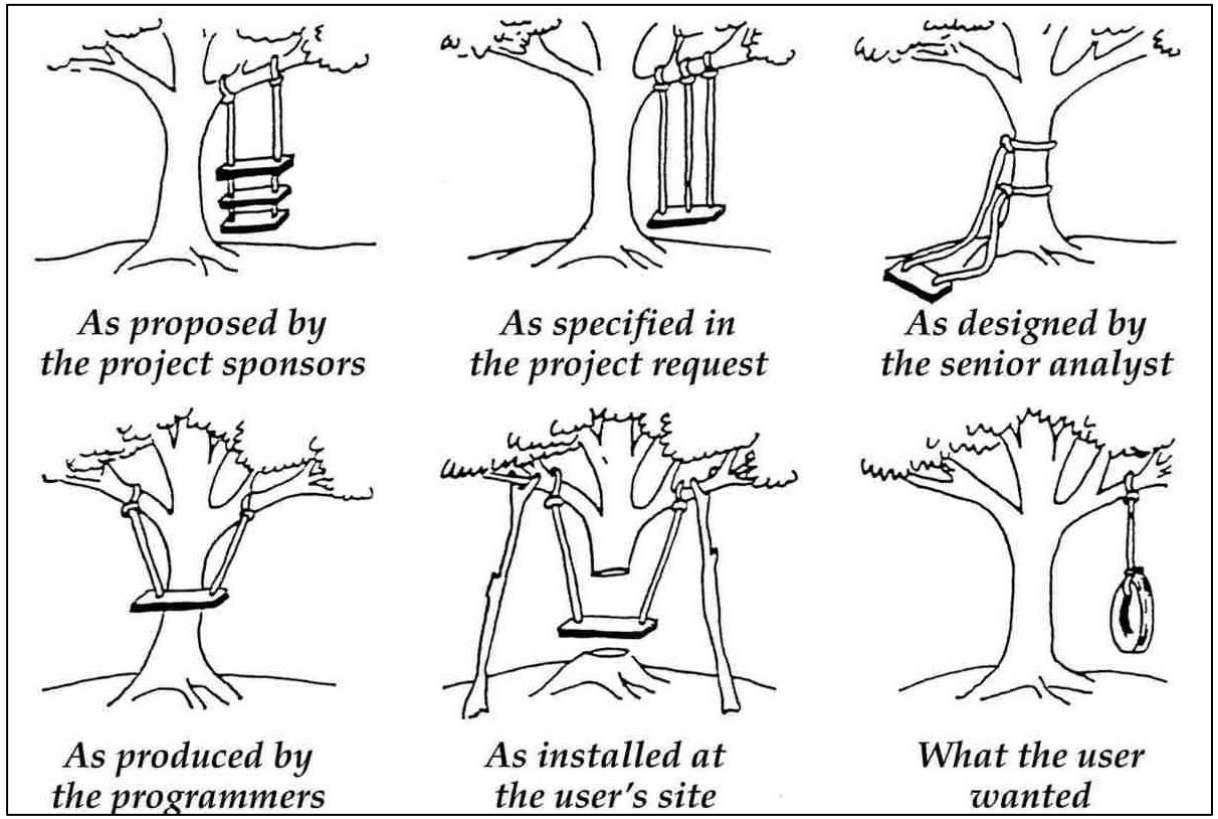
Requirement Concepts
Requirement Definition 1/3

- What is requirement?

  - A statement of a **service** the system must do OR
  - A statement of a **constraint** the system must satisfy

Requirement Concepts
Requirement Definition 2/3

- Why do we need requirement definition?



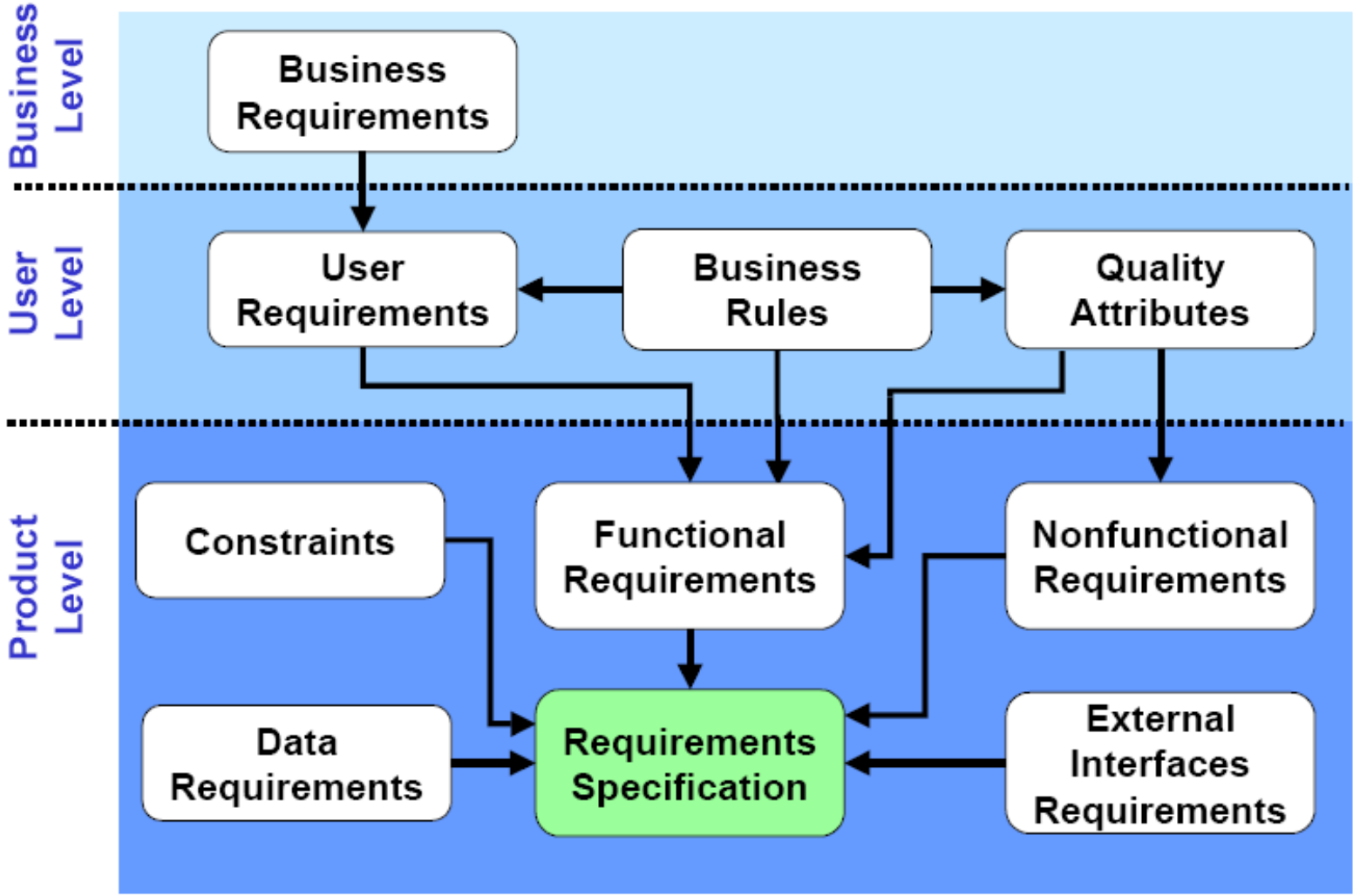
(Tire swing picture from 1970s)

Requirement Concepts
Requirement Definition 3/3

- Purpose of requirement:
  - Requirements often serve as:
    - The basis for a bid for a contract - therefore must be high-level to open for interpretation
    - The basis for the contract itself - therefore must be detailed
  - Thus, requirements can be high-level or detailed
- What are not Requirements:
  - Design or implementation details (other than known constraints)
  - Project planning information
  - Testing information

UNIVERSITY of GREENWICH
Alliance with FPT Education

Requirement Concepts
Requirements Classification 1/5



UNIVERSITY of GREENWICH
Alliance with FPT Education

Requirement Concepts
Requirements Classification 2/5

- Requirement may be classified as
  - Functional
    - A service the system has to perform
    - May include information the system must contain
  - Non-functional
    - A constraints the system must satisfy
- "Functional requirements deal with the What, non functional requirements deal with the How" *(Ariel Schlesinger)*[1]

Requirement Concepts
Requirements Classification 3/5

- Sample of functional requirements
  - The “Data Entry Module” should provide the following functionality:
    - Data Entry for HR: allows HR staff to enter payroll data and the like, either via web-based forms or by importing data from Excel files
    - Data Entry for Regional offices: allows the Regional offices to enter billing data, either via web-based forms or by importing data from Excel files

UNIVERSITY of GREENWICH
Alliance with FPT Education

Requirement Concepts
Requirements Classification 4/5

- Sample of non-functional requirements
  - –Product requirements
    - Requirements which specify that the delivered product must behave in a particular way
    - Categories: performance, reliability, usability, security, cultural,…
  - –Organisational requirements
    - Requirements which are a consequence of organisational policies and procedures
    - Categories: technology, process, operation, time, budget,…

Requirement Concepts
Requirements Classification 5/5

- Sample of non-functional requirements
  - External requirements
    - Requirements which arise from factors which are external to the system and its development process.
    - Categories: interoperability requirements, legislative requirements,…

Requirement Modeling
Modeling objectives

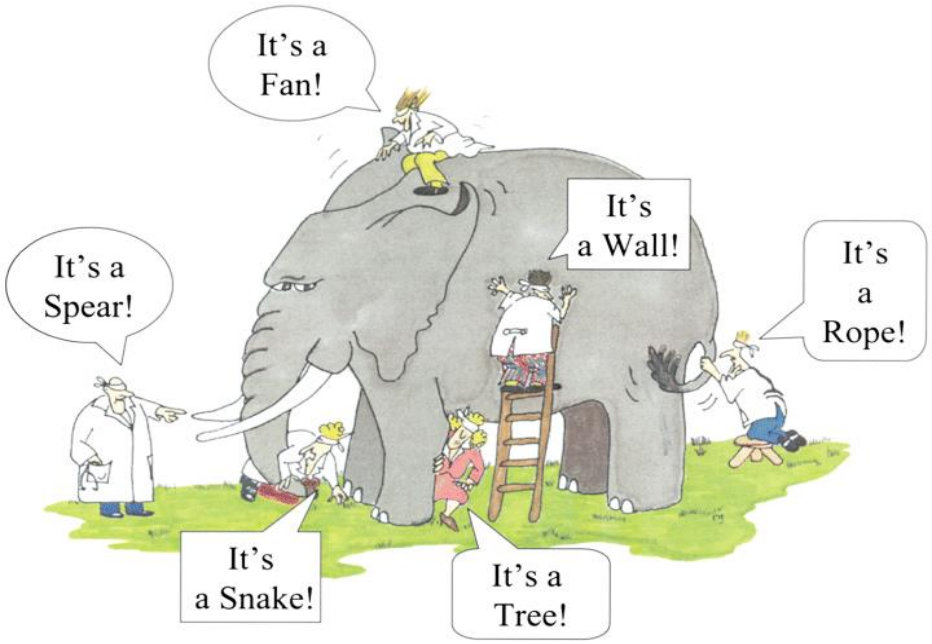
- Why requirement modeling?

To understand clearly the functionalities of system

To present the system from different perspectives

Requirement Modeling
System Modeling 1/3

- What is System Models?
  - The system models presents an abstract software aspects, which helps understan requirements in block diagram form, and required software elements & tasks.
- Models help present different aspects different abstraction



UNIVERSITY of GREENWICH
Alliance with FPT Education

Requirement Modeling
System Modeling 2/3

- Object Oriented Analysis and Design (OOAD):
  - **Dynamic (or Behavioral) Model**: emphasizes the dynamic behavior of the system by showing collaborations among objects and changes to the internal states of objects.
  - **Static (or Structural) Model**: emphasizes the static structure of the system using objects, attributes, operations and relationships.

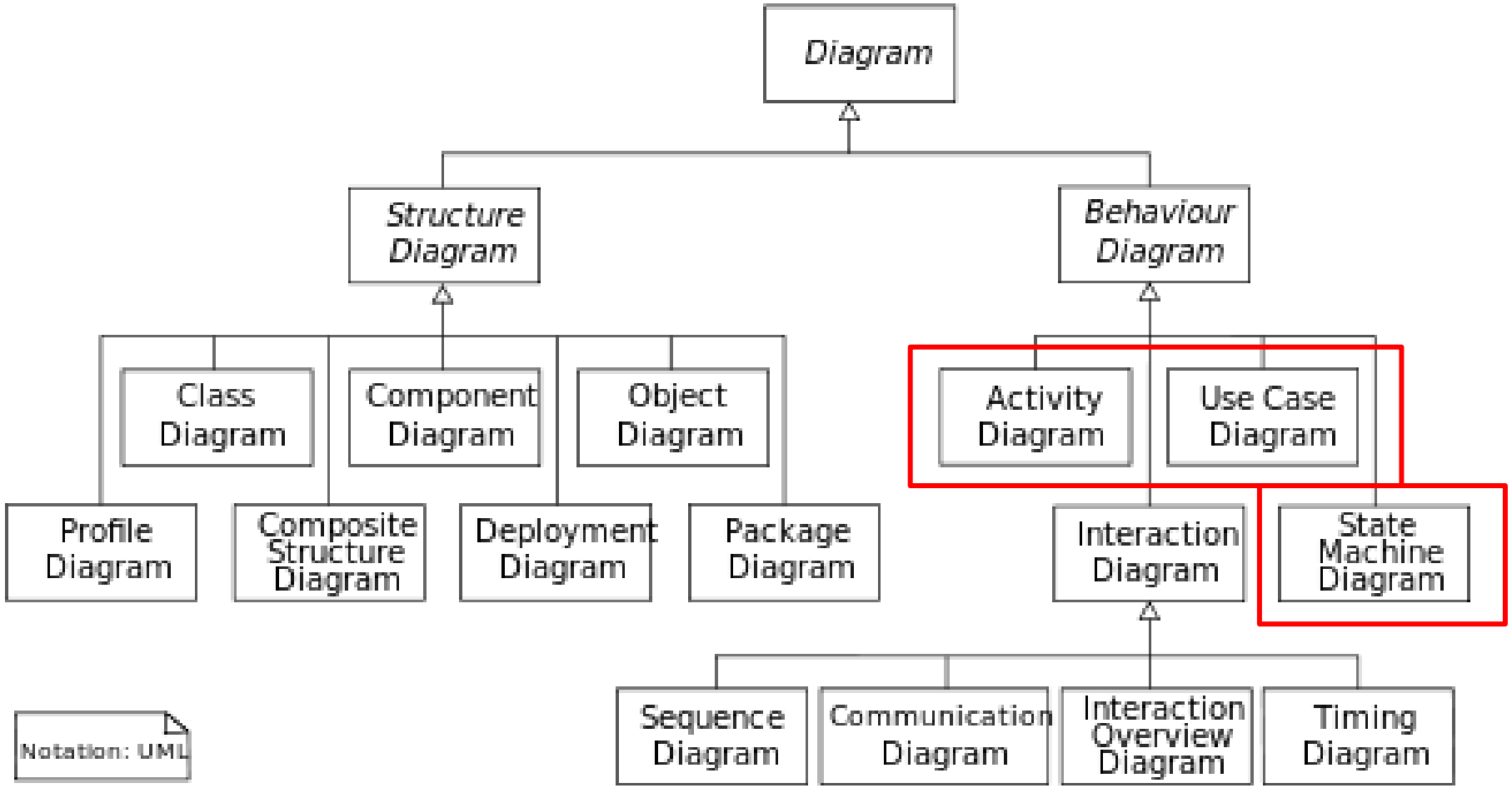
UNIVERSITY of GREENWICH
Alliance with FPT Education

Requirement Modeling
System Modeling 3/3

- Structured systems analysis and design method (SSADM):
  - **Logical data model**: the data requirements of the system being designed. A data model containing entities, attributes and relationships. (Entity Relationship Diagram – ERD)
  - **Data Flow Diagram (DFD)**: how data moves around an information system.
  - **Entity Event Model**:
    - **Entity Behavior Model**: the events that affect each entity and the sequence.
    - **Event Model**: designing for each event the process to coordinate entity life histories.

Requirement Modeling
System Modeling in OOAD

- Common requirement modeling tools (UML):



UNIVERSITY of GREENWICH
Alliance with FPT Education

Modeling Tools - Use Case
Use Case Concepts 1/2

- Requirements capture
  - –Requirements are reason-for-existence of any software development project
  - –Defines and delineates user-requirements
    - Defines the functionality to be provided
    - Identifies the goals to be achieved
  - –Must be precisely and completely understood
  - –Requirements often changes, thus must be well-documented

UNIVERSITY of GREENWICH
Alliance with FPT Education

Modeling Tools - Use Case
Use Case Concepts 2/2

- Requirements capture with UML
  - Use Case diagram
    - Shows a set of use cases, actors and their relationships
  - Captures problem-domain in terms of:
    - functionality to be provided (Use Cases)
    - the “roles” (Actors) for whom these functions are performed

Modeling Tools - Use Case
Use Case Diagram - Notations

Subject/System boundary:

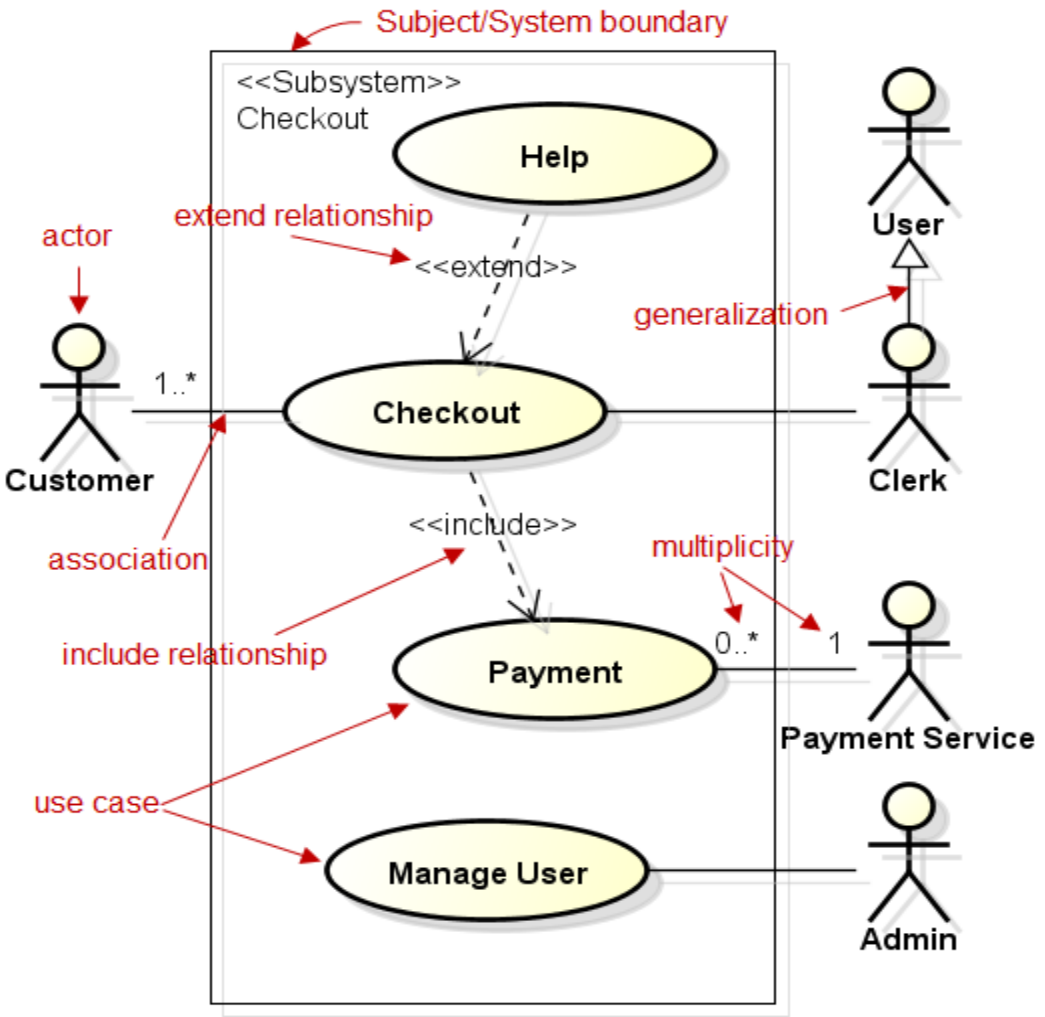
Defines and represents boundaries of a business, software system, physical system or device, subsystem, component or even single class in relation to the requirements gathering and analysis.

Actor:

Specifies a role played by an external entity that interacts with the subject, user of designed system, other system/hardware using services of the subject.

Use case:

Capture requirements of systems, describe functionality provided by those systems, and determine the requirements the systems pose on their environment.



Modeling Tools - Use Case
Use Case Diagram - Notations

Association:

An association between an actor and a use case indicates that the actor and the use case somehow interact or communicate with each other.

Extend relationship:

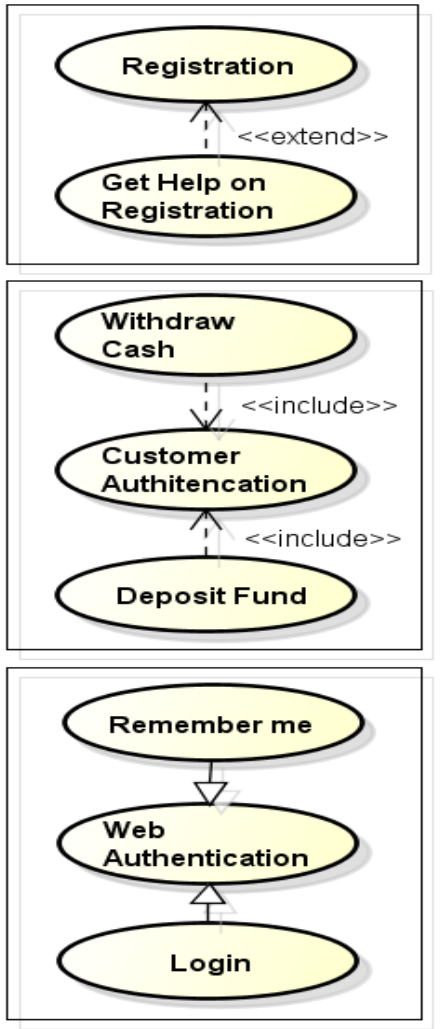
A directed relationship specifies that one use case (extension) extends the behavior of another use case (base). Extension use case is meaningful on its own, it is independent of the base use case.

Include relationship:

A directed relationship in which one use case (the base use case) includes the functionality of another use case (the inclusion use case). The include relationship supports the reuse of functionality in a use-case model. The inclusion use case cannot stand alone and the base use case is not complete without the inclusion one.

Generalization relationship:

A taxonomic relationship in which one actor/use case (the child) is based on another actor/use case (the parent). The child actor/use case inherits the features of the parent.



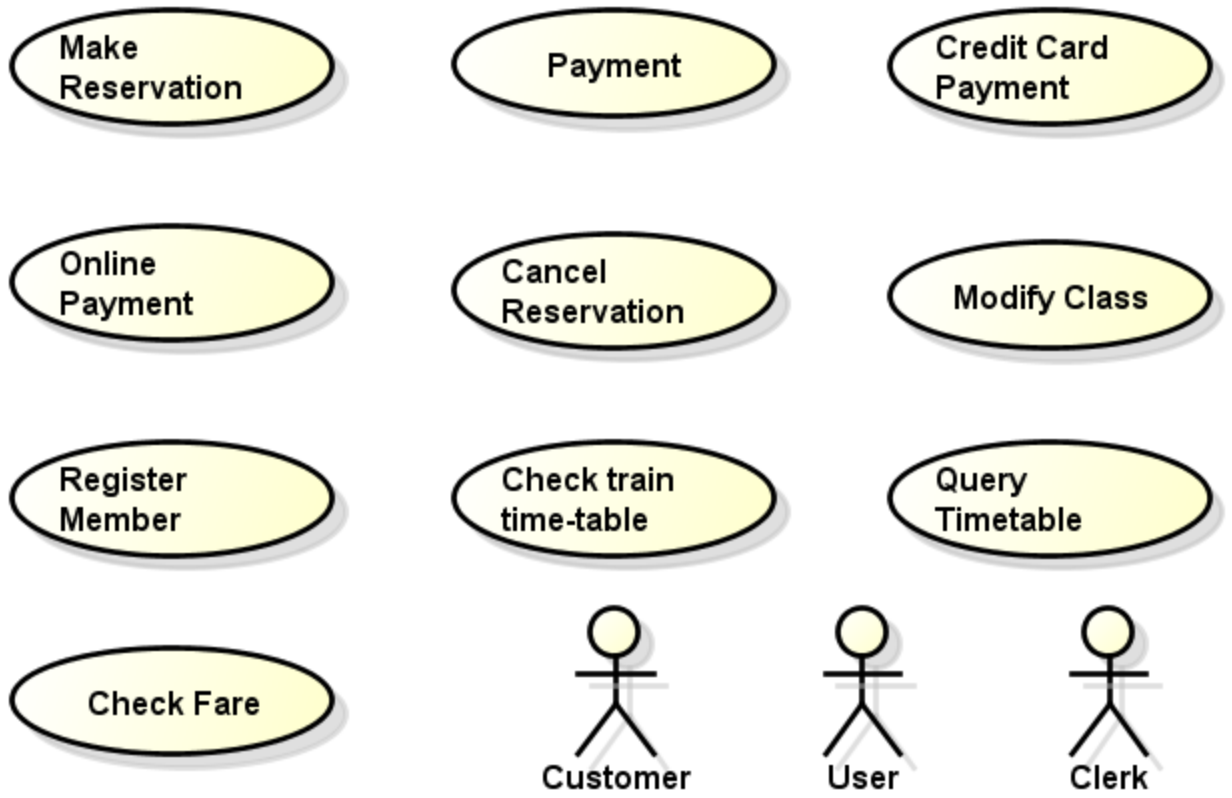
Modeling Tools - Use Case
Use Case – Example 1/3

–A company wants to develop a ticketing and reservation system. This must support advance booking of tickets, cancellation of tickets and **change of class** of a ticket. All these are handled by a Reservation Clerk.

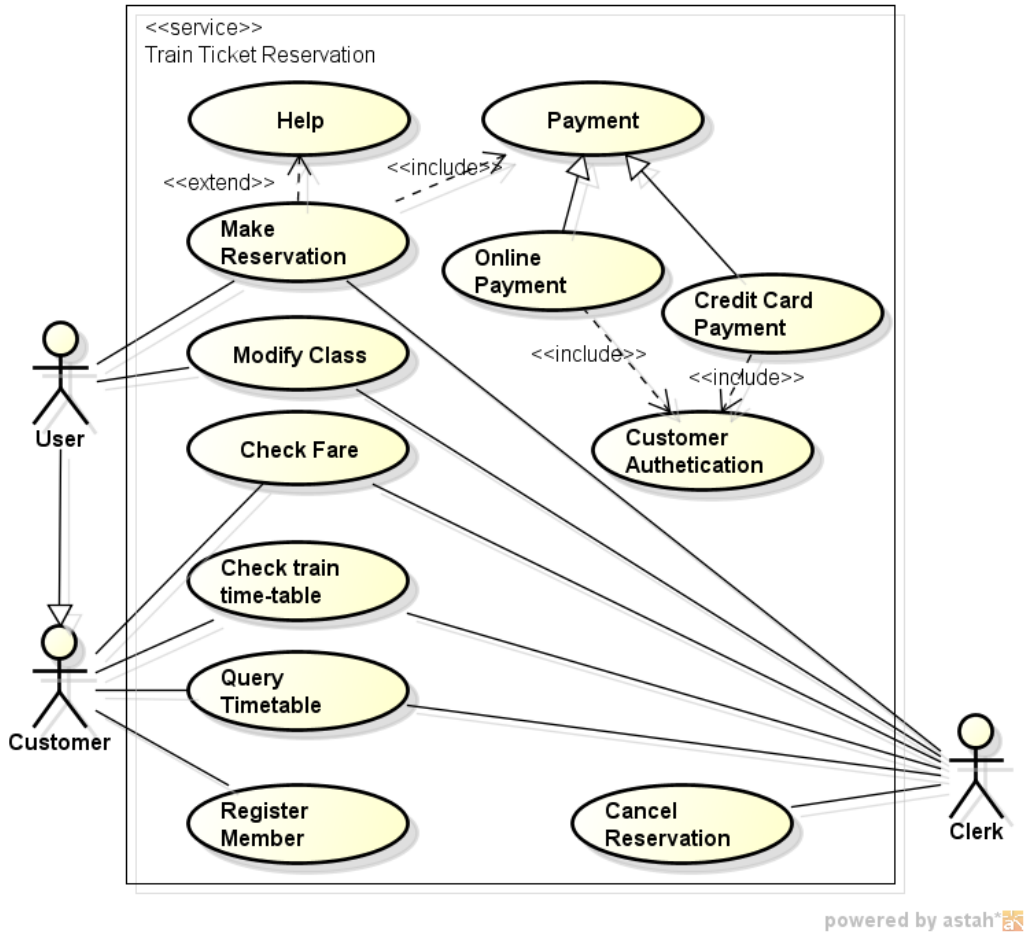
–The system will also have a Web site where users can **register** themselves and purchase tickets online. They can **pay** either by using their **online banking account** or by **credit card**. **Reservations** made over the internet can only be **cancelled** across the counter.

–The system will also have a querying facility that allows users to **check train time-tables**, **fares** and **availability of tickets**.

Modeling Tools - Use Case
Use Case – Example 2/3



Modeling Tools - Use Case
Use Case Diagram – Example 3/3



UNIVERSITY of GREENWICH
Alliance with FPT Education

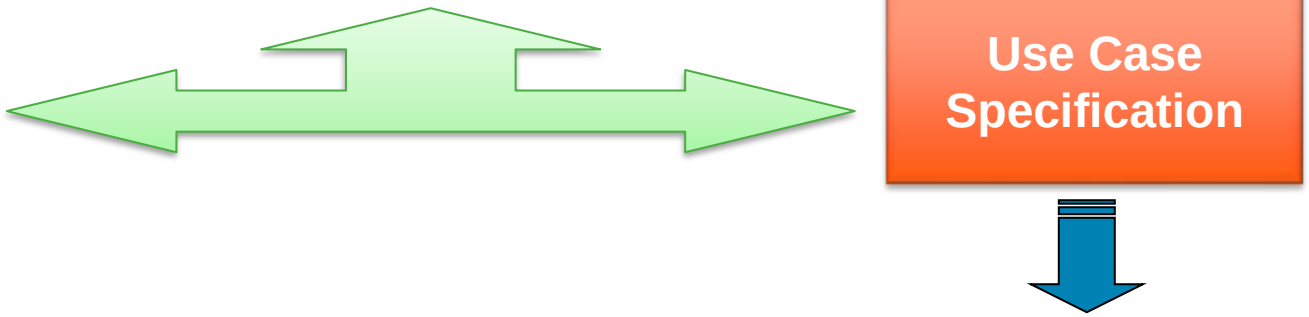
Modeling Tools - Use Case
Use Case Specification 1/6

- Text description of use case functionality in the user language and terminology
- No specific UML format
- Describes WHAT and not HOW
- Typically includes:
  - Objectives of the use case
  - How the use case is initiated
  - The flow of events (main flow, alternative flow)
  - How the use case finishes with a value to the actor and more...

Modeling Tools - Use Case
Use Case Specification 2/6

- Use case description serves as a ‘bridge’ between stakeholders of a system and the development team.

Systems analyst produce use case diagram & use case specification in consultation with end users



Modeling Tools - Use Case
Use Case Specification 3/6

- Flow of events
  - Use Case is an abstraction of behavior (set of sequences)
  - The behavior of the Use Case can be described by a “flow of events” - which spells out in detail what exactly the Use Case does
    - main flow: what happens and in what order when all is well
    - alternate flow(s): what happens and in what order when something goes wrong
    - exception flow: things don’t always go as planned. An exception is an error condition that is important enough to the application to capture

UNIVERSITY of GREENWICH
Alliance with FPT Education

Modeling Tools - Use Case

Use Case Specification 4/6

Key components	Explanation
Name	*Clear, unique name of the use case (verb, goal-driven)*
Actors	*Someone or something that interacts with the use case*
Description	*Brief overview of the use case, describing the main idea*
Goal	*What the actors achieve with this use case*
Pre-condition	*State(s) the system can be in before the use case starts*
Trigger	*Event that causes the use case to be initiated*
Post-condition	*State(s) the system can be in after the use case finishes*
Normal flow	*Typical (primary) processing path*
Alternative flow	*Alternative (secondary) processing path*
Exception flow	*When things go wrong at the system level*
Others	Business rules, Assumption, Notes, etc.

UNIVERSITY of GREENWICH
Alliance with FPT Education

Modeling Tools - Use Case
Use Case Specification 5/6

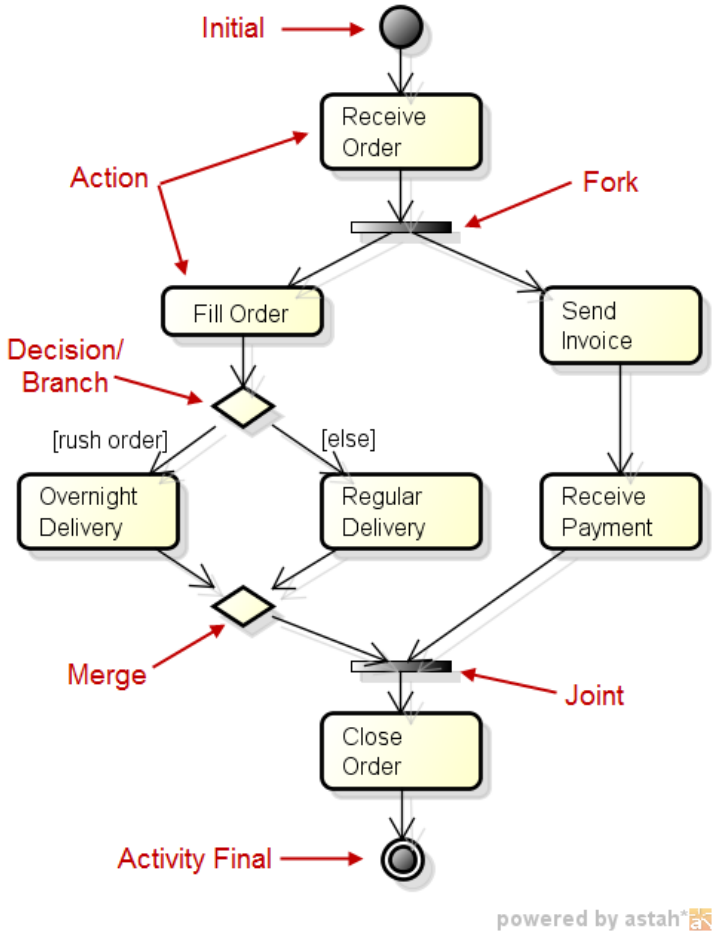
Make a seat reservation use case	
Name	*Make reservation*
Actors	*Passenger*
Description	*Allows a passenger to book a plane seat for a journey from the Website*
Goal	*Reserve a seat*
Pre-condition	*Main Webpage is displayed successfully*
Trigger	*User clicks on "Reserve seat" button on the main Webpage*
Post-condition	• *A seat is booked* • *Number of available seats is reduced*

UNIVERSITY of GREENWICH
Alliance with FPT Education

Modeling Tools - Use Case
Use Case Specification 6/6

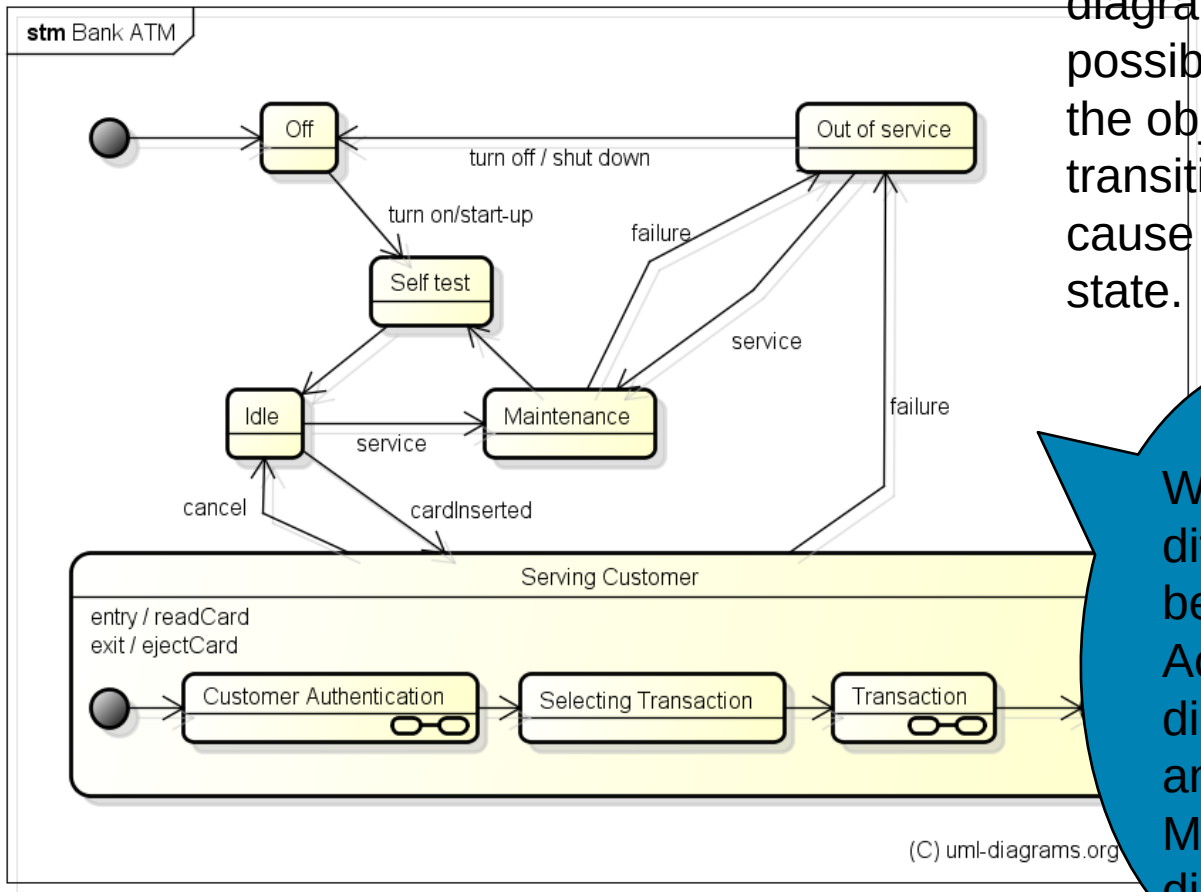
Make a seat reservation use case	
Normal flow	*[User log in and reserve a seat successfully]* 1. User logs in 2. User specifies a flight and travel details 3. User specifies passenger details 4. User specifies payment details 5. User confirms transaction
Alternative flow	*[When no seat is available on the selected date]* • Show option to select another day • Repeat steps in normal flow
Exception flow	*[When a payment is failed]* • Notify error with the payment • Give an option to re-enter payment details or other payment method

Modeling Tools - Activities Diagram Definition and Notations



- Activity diagram
  - Describes the workflow behaviour of a system including a sequence of activities performed from start to finish
  - Activities could be performed:
    - sequential order
    - parallel
    - conditional transition

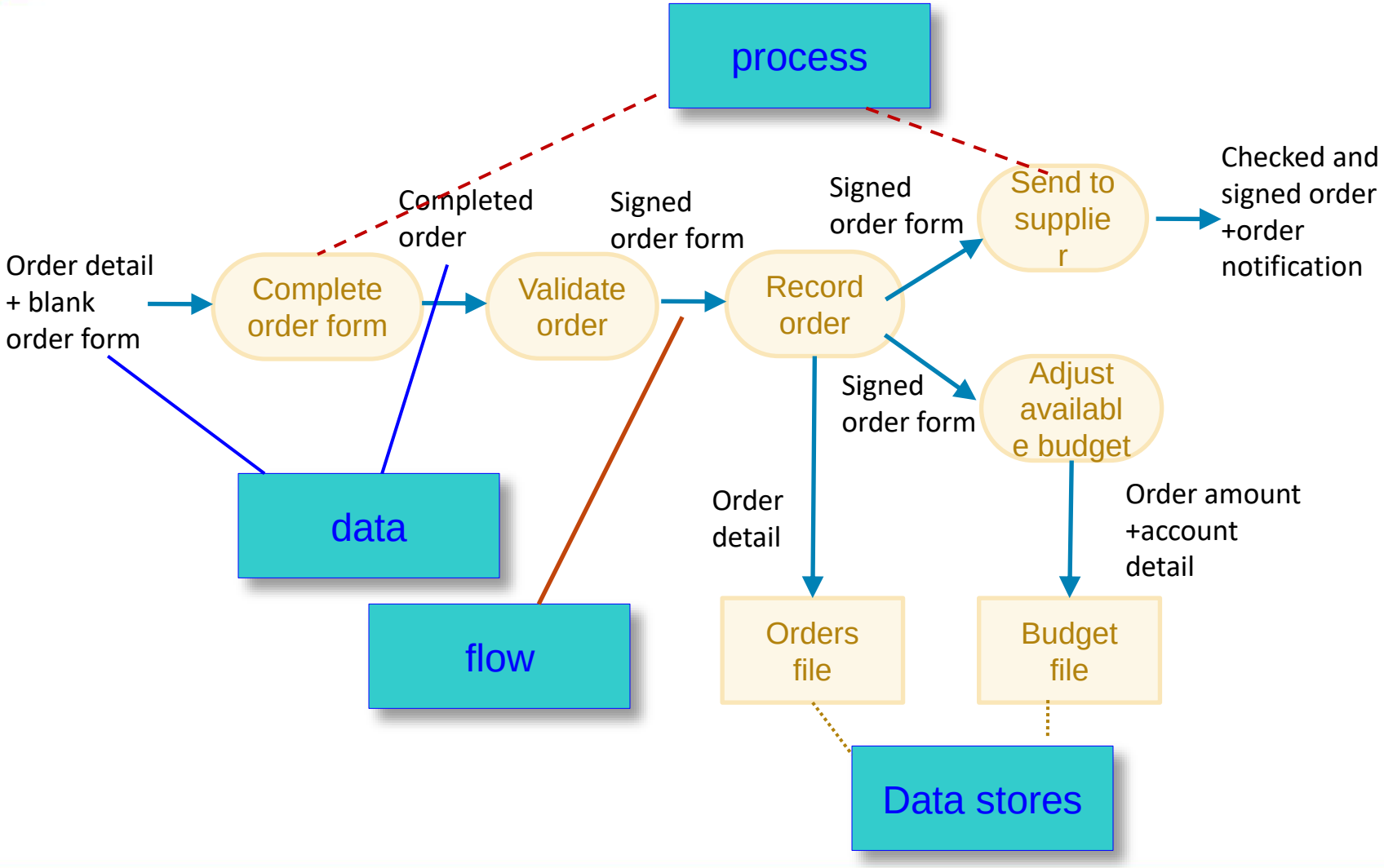
Modeling Tools - State Machine Definition and Notations



A State Machine diagram shows the possible states of the object and the transitions that cause a change in state.

What is different between Activity diagram and State Machine diagram?

Modeling Tools – DFD Sample and Notations



Modeling Tools – ERD Sample and Notations

- Concept → Logical → Physical Data Diagram