

---

## **Software Construction and User Interfaces (SE/ComS 319)**

Ali Jannesari

Department of Computer Science

Iowa State University, Spring 2019

# **GUIDELINES FOR SYSTEM MODELING – GROUP PROJECT**

# Use case diagrams (UML)

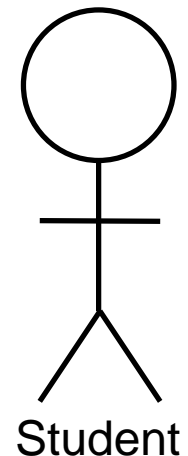
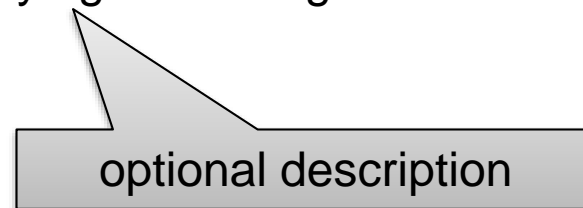
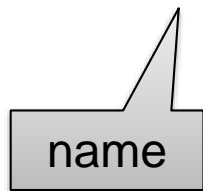
---

- Use case diagrams are used during the requirement engineering to represent the externally visible behavior of the system
- An actor specifies a role of a user or other system that interacts with the system we are analyzing
- A use case represents a class of functions offered by the system
- A use case model is the set of all use cases that describe the entire functionality of the system
- A use case diagram includes
  - Actors, use cases, associations, system boundary

# Actors

---

- An actor is a model for an external entity that interacts with the system:
  - Administrator, end user, environment, external systems, ...
- An actor has a unique name and optionally a description
- Example:
  - **Student:** A person who is studying or training at a university or college
  - **Random number generator**



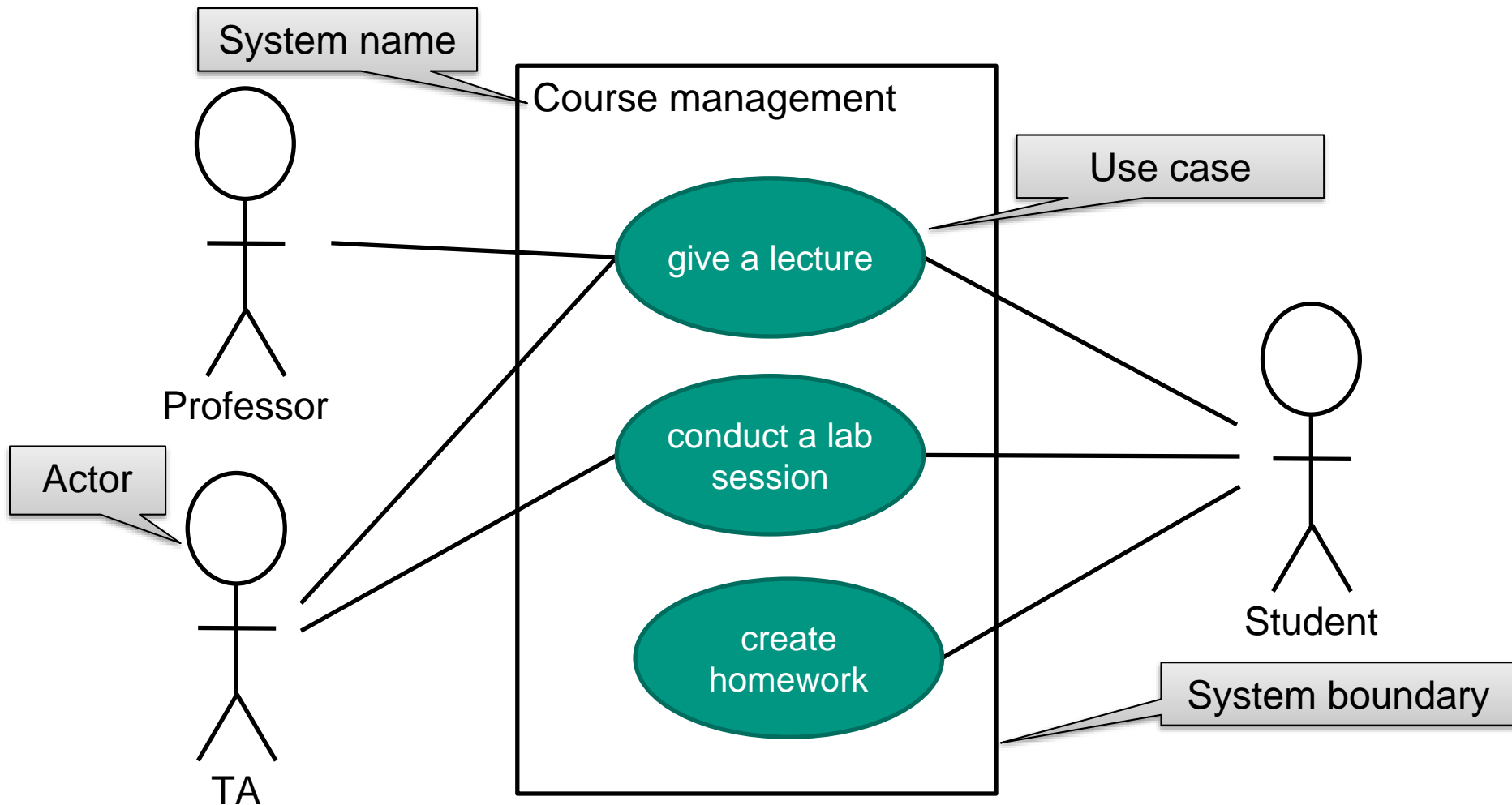
# Use cases – How to describe them?

---

- Use cases can be described with text, with a focus on the interaction between actor and system
- The description of a use case with text consists of 6 parts:
  - Unique name
  - Participating actors
  - Input actions
  - Output actions
  - Event flow
  - Special requirements
- They also can be described with activity diagram



# Use case diagram – Example



# Scenarios

---

- A scenario...
  - ... is the description of an event or sequence of actions and events
  - ... is the description of how to use a textual system from **a user's perspective**
  - ... can contain text, images, videos, and schedules, as well as details about the workplace, the social environment, and resource constraints

## Scenarios – Example "Burning warehouse"

---

- While Bob drives his main car along the main road, he notices smoke rising from a warehouse. His colleague, Alice, reports the emergency from the vehicle.
  - Alice enters the address of the warehouse into her mobile computer, a brief description of the location (e.g., north-west corner) and a priority.
  - She confirms her entry and waits for a confirmation.
  - John, the dispatcher at the control room, is alerted to the emergency by a beep on his computer. He analyzes the information Alice sent him and confirms the message. He alerts the fire department and passes the expected time of arrival to Alice.
  - Alice receives the confirmation and expected arrival time.
-

# Use case from scenario

---

- Find all use cases in the example scenario that all instances specify how to report a fire
  - Example: "**Report Emergency**" in the first paragraph of the scenarios is a candidate for a use case
- Describe each of these use cases **as accurately as possible**:
  - Participating actors
  - Describe their input actions
  - Describe their event flow
  - Describe their output actions
  - Describe exceptions
  - Describe non-functional requirements



## Use case: "Report emergency" (1)

---

- Name of use case: Report emergency
- Participating actors:
  - Policeman (Bob and Alice in this scenario)
  - Dispatcher (John in this scenario)
- Exceptions:
  - The police officer will be notified immediately if the connection between the terminal and the headquarters breaks.
  - The dispatcher is notified immediately when the connection between a police officer and the headquarters breaks.

## Use case: "Report emergency" (2)

---

- Event flow:
  - The policeman activates the "Report Emergency" feature on his terminal. FRIEND system [an external system] responds by displaying a form.
  - The police officer completes the form by entering the emergency level, the type of assignment, the address, and a brief description of the situation. The policeman also describes a reaction to the emergency situation.
  - The dispatcher creates an incident in the database by calling the "Open Incident" use case. He chooses a reaction and confirms the message.
  - The policeman receives the confirmation and chooses the reaction.
- Nonfunctional requirements:
  - The police report will be confirmed within 30 seconds. The answer arrives at the police no later than 30 seconds after being sent by the dispatcher.

## Use case: "Request resources" (1)

---

- Actors:
  - Operations Manager: The person responsible for the deployment
  - Resource Requester: Responsible for requesting and releasing resources managed by the FRIEND system.
  - Dispatcher: enters incidents, updates and deletes incidents in the system. He is also responsible for closing incidents.
  - Policeman: Reports incidents

## Use case: "Request resources" (2)

---

- Name of Use Case: Request resources
- Participating actors:
  - Policeman (Bob and Alice in this scenario)
  - Dispatcher (John in this scenario)
  - Resource requester
  - Operations manager
- Input actions:
  - The resource requester has selected an available resource
- Flow of events:
  - The resource requester chooses an incident
  - The resource is assigned to the incident

## Use case: "Request resources" (3)

---

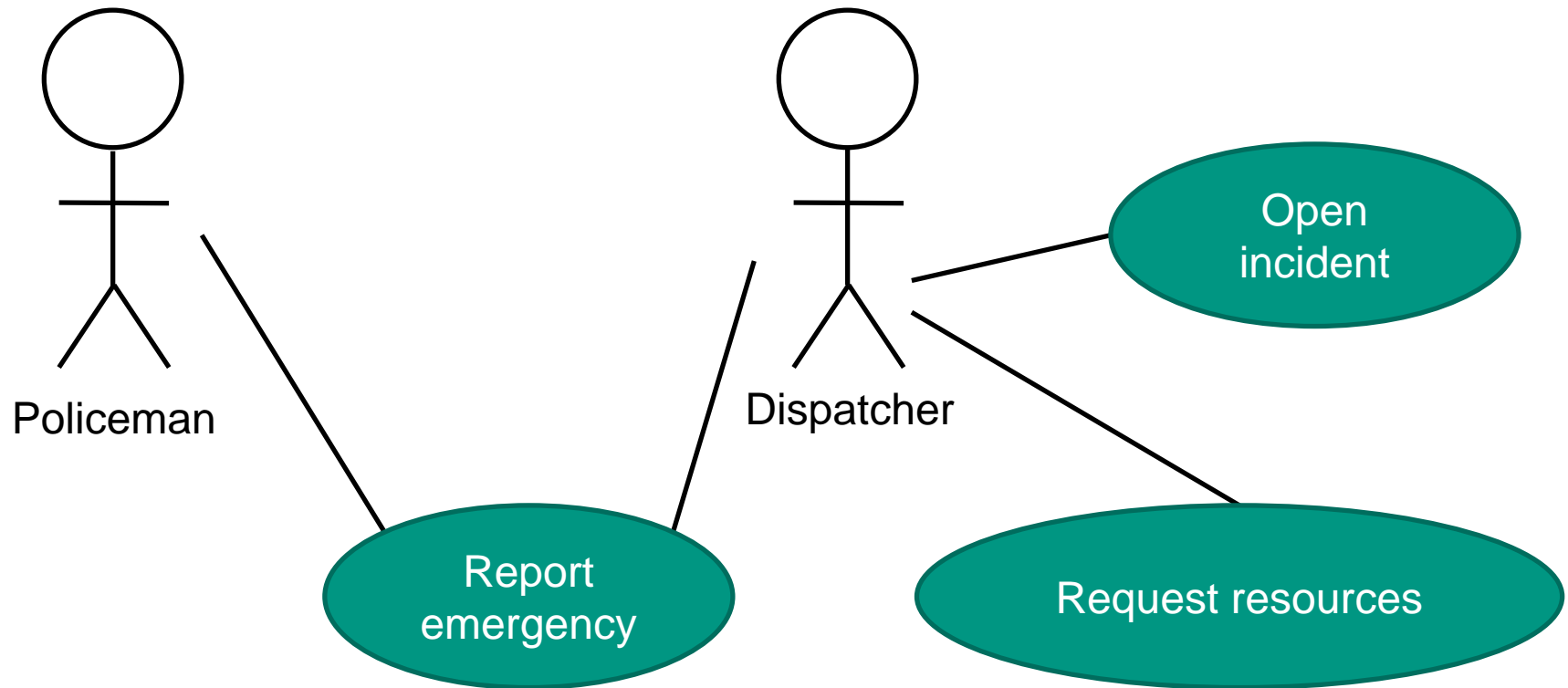
- Output actions:
  - The use case is ready when the resource has been assigned.
  - The selected resource is not available for other requests.
- Special requirements:
  - The Operations Manager is responsible for the use of resources

# How to formulate use cases?

---

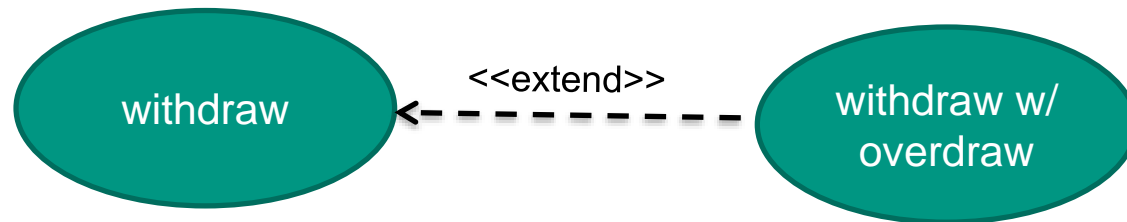
- Name of the use case:
  - For example: Report emergency
- Find the actors:
  - Generalize the concrete names ("Bob") to participating actors ("Policeman")
  - Participating actors:
    - Policeman (Bob and Alice in the example scenario)
    - Dispatcher (John in the example scenario)
- Find the event flow:
  - Described in natural language

# Use case model – Report emergency



## <<extend>> Relationship

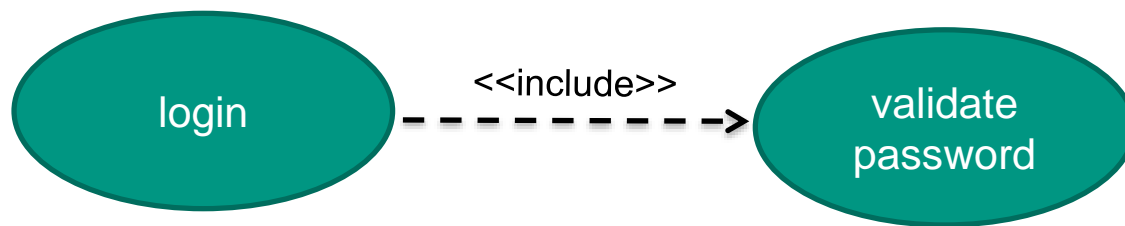
- Use cases can be related (associated) to each other.
- **Extend** relationship represents rarely-called use cases or exceptional functionality.
  - A relationship between one use case which is extended by some optional use case (added features).
  - For example, use withdraw money can be extended by use case process overdraft





## <<include>> Relationship

- **Include** relationship represents functionality that is used by more than one use case.
  - A relationship between one use case which requires the existence of another use case, and the latter, in isolation, is not meaningful to the user.
  - For example, validate password use case is included in login use case



# Use case text – Describing use case in text format

---

- Use case name
- Main **scenario**
  - **Steps**
- Extensions
  - Extension condition; steps
- Specify **what** to do, not **how** to do
- Do not specify user interface
- Optional: priority, trigger, pre-condition, post-condition (guarantees), sub-use case

# Use case text – Example

---

- **Name:**
  - Create homework
- **Participating actor:**
  - College student
- **Input condition:**
  - Student receives exercise sheet
  - Student is healthy
- **Output condition:**
  - Student makes solution
- **Flow of events:**
  - Student brings current exercise sheet
  - Student reads through the tasks
  - Student solves the task and enters it into the computer
  - Student prints the solution
  - Student submit the solution
- **Special requirements:**
  - No

# Activity diagrams

---

- An activity – multiple actions
  - Can be used to describe a use case
  - Can represent parallel relationship
- An activity diagram describes a procedure
  - Operational or business processes
  - Technical processes of workflows and use cases
  - Concrete algorithmic processes in programs
- Activity diagrams consist of
  - Action, object nodes and control nodes, as well
  - Object flows and control flows.

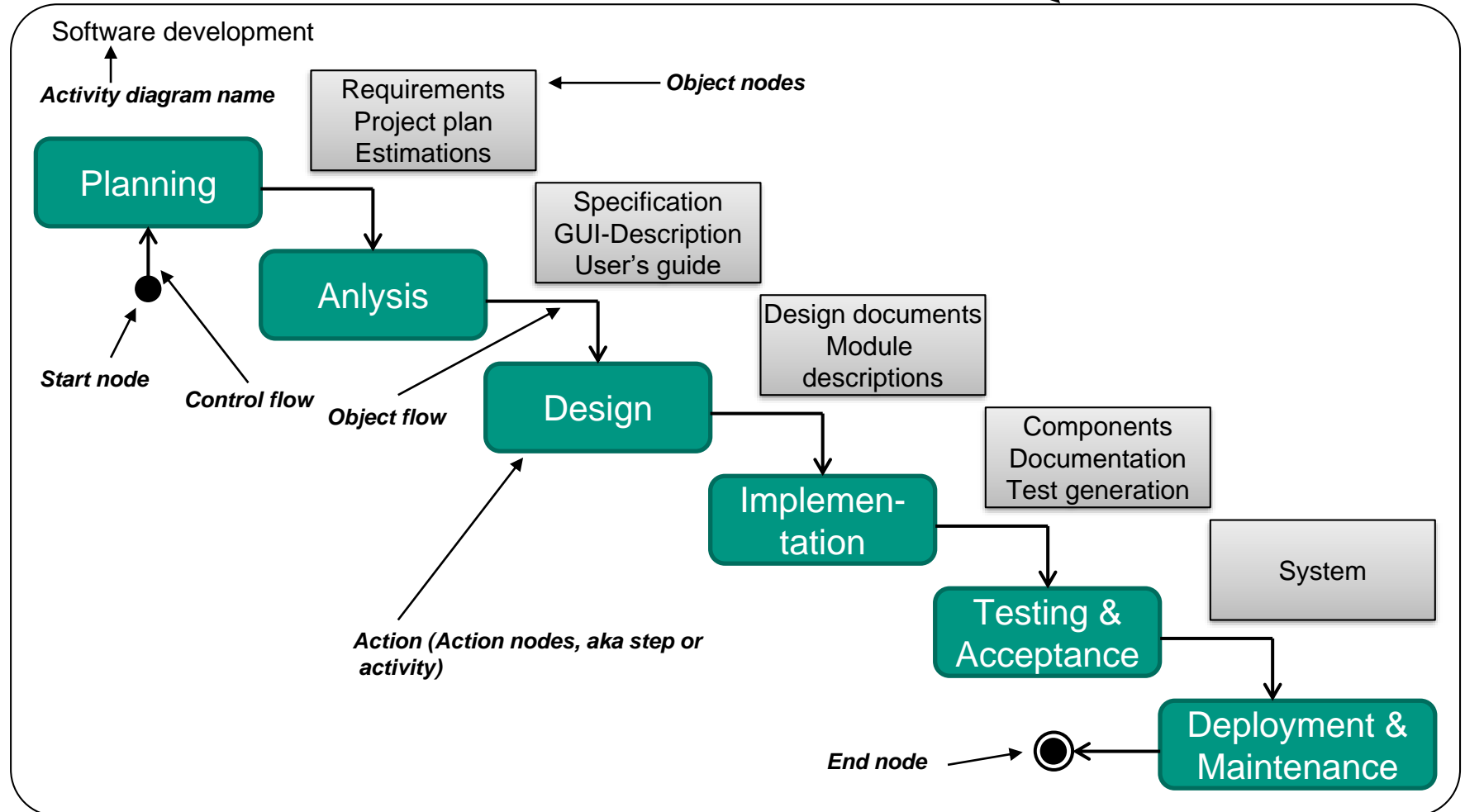
# Activity diagram – Main components

---

- Main components
  - Start
  - Actions
  - Fork/Join
  - Decision/Merge
  - Flow
  - Final

# Example – Waterfall software process model

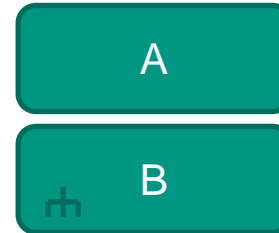
Activity (also: Activity diagram)



# Activity diagram symbols and elements (1)

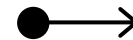
- Actions

- Elementary action
- Nested action



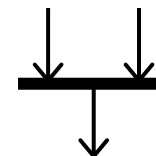
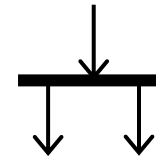
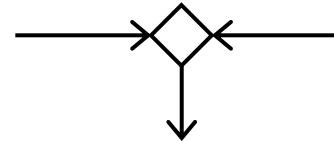
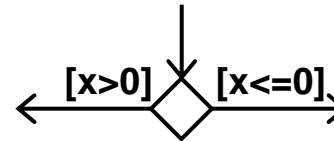
- Nodes

- Starting node
  - Starting point of a process
- End nodes
  - Ends all actions and control flows
- Flow final
  - Ends a single object flow and control flow



## Activity diagram symbols and elements (2)

- Decision
  - Conditional branching
- Merging
  - "or" connecting
- Forking
  - Dividing a control flow
- Synchronization
  - "and" joining



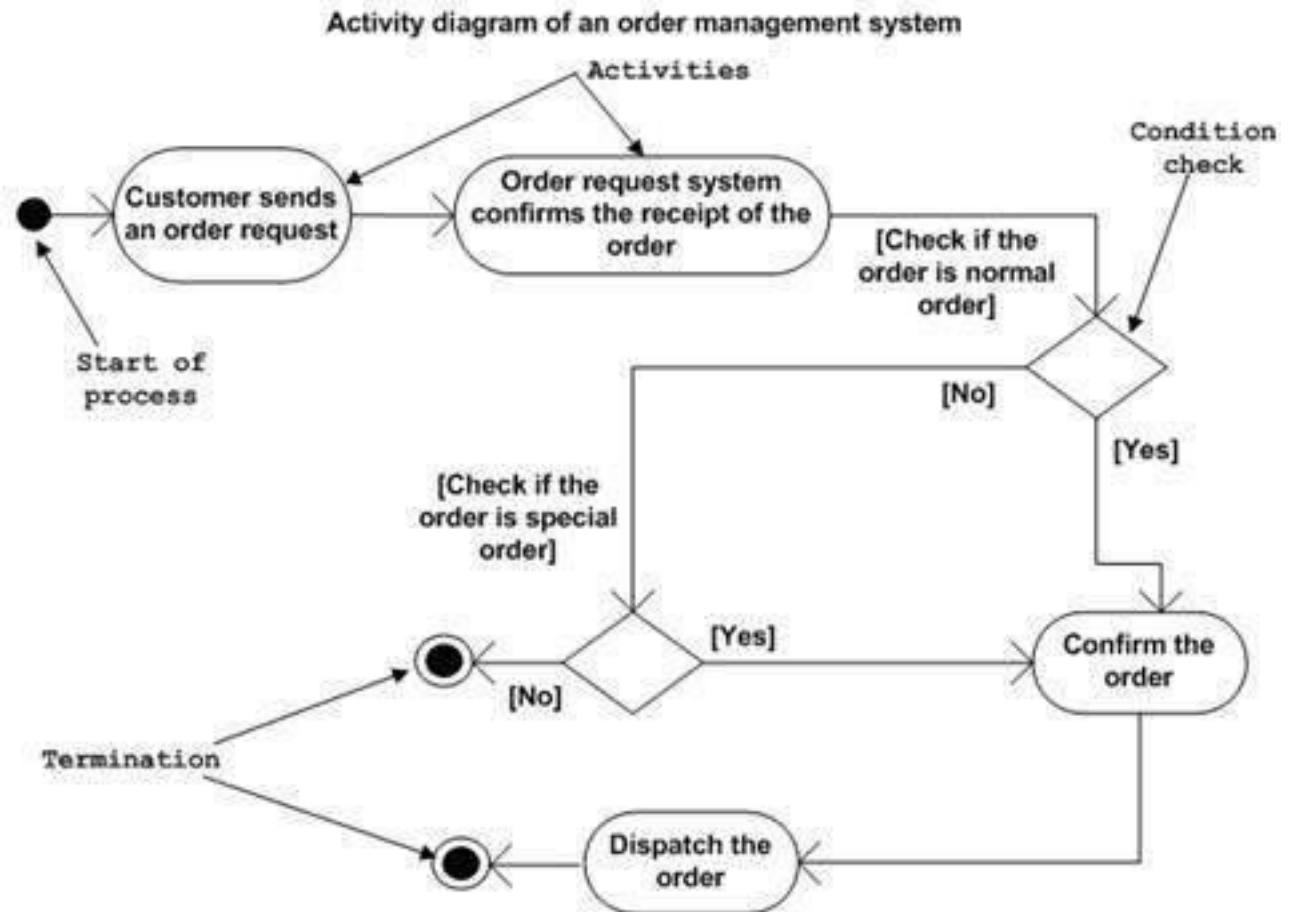


# Activity diagram – Example: Order management

- An activity diagram for order processing

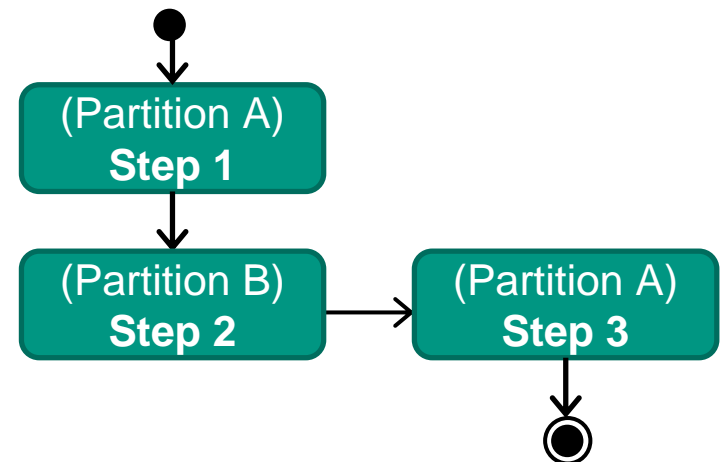
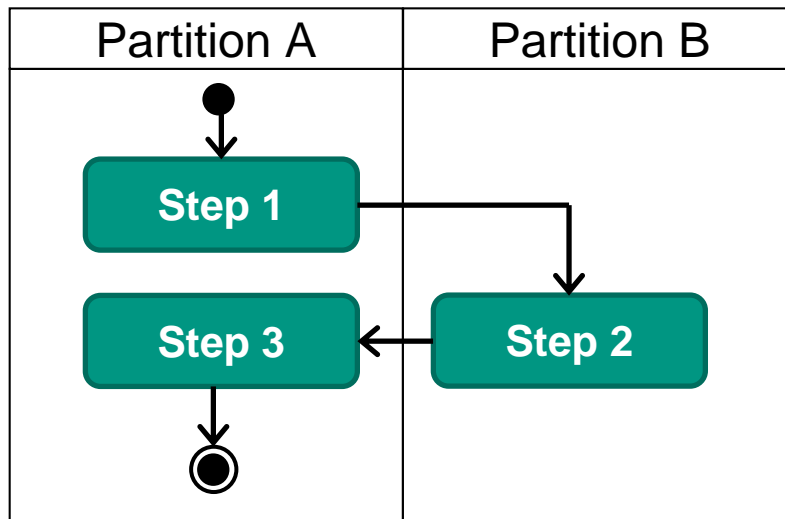
Source:

<https://www.tutorialspoint.com/uml/>



## Activity diagram symbols and elements (5)

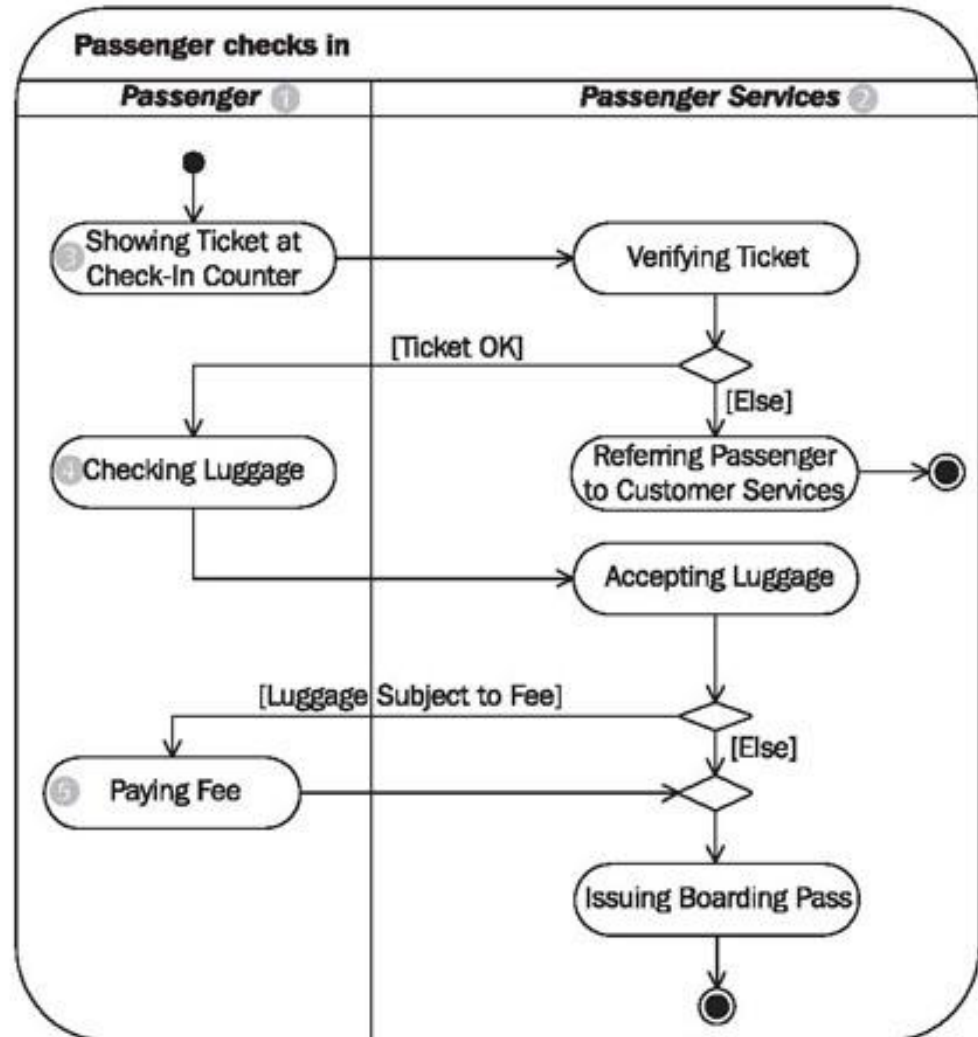
- Partitions (areas of responsibility)
  - Partitions describe who or what is responsible for a node or what common feature characterizes it.
  - For example, partitions could be different computers working together (e.g. server and client)



# Activity diagram – Example with partitions

- An activity diagram with partitions

Source: <https://sourcemaking.com/uml/>



# Summary

---

- Use cases (UML)
- Use case text
- Activity diagram