

The background of the slide is a light gray gradient. It is decorated with numerous realistic water droplets of various sizes. Some droplets are large and prominent, while others are small and subtle. They are scattered across the slide, with a higher concentration in the top-left and bottom-right corners. The droplets have highlights and shadows, giving them a three-dimensional appearance.

WEEK 4 LECTURE: UML ACTIVITY AND STATE DIAGRAMS

STATE AND ACTIVITY DIAGRAMS

- State machine diagrams show the behavior of the class/subsystem in response to external stimuli.
- Activity diagrams show the behavior of the the class/subsystem in response to internal processing.

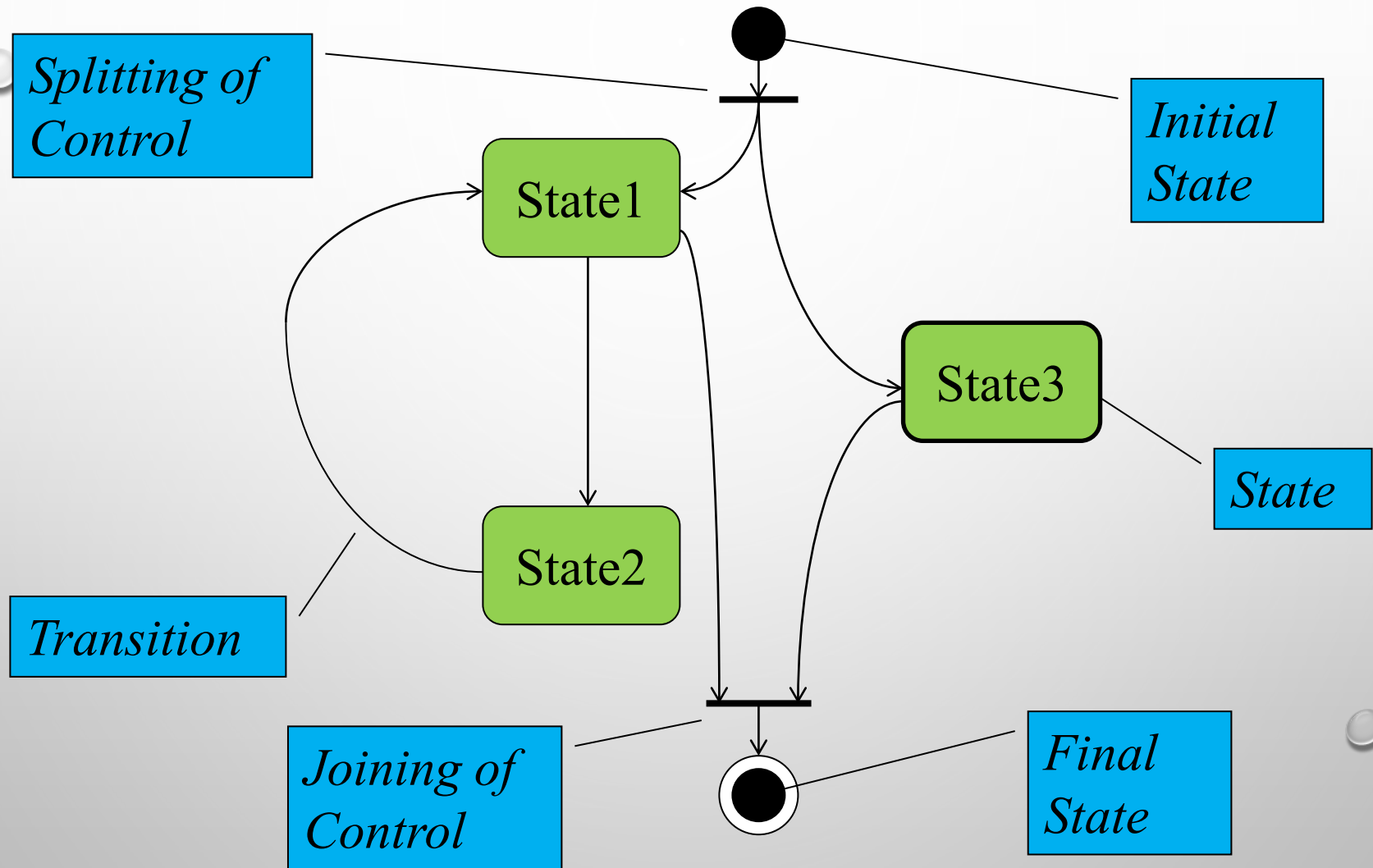
STATECHART DIAGRAMS

- Useful for describing the behaviour of individual objects over the full set of use cases that affect those objects
- State chart diagrams describe
 - All of the states that an object can have,
 - The events under which an object changes state (transitions),
 - The conditions that must be fulfilled before the transition will occur (guards),
 - The activities undertaken during the life of an object (actions).

STATE ANALYSIS NOTATION

- Describes the states an object can have during its lifecycle, the behaviour in those states, and the events that can cause the state to change.
- States—representing the distinct behaviours of the class or subsystem. Often represented by state objects.
- Transitions—the processes by which the class or subsystem changes behaviour.

SAMPLE STATE MACHINE DIAGRAM



ELEMENTS OF STATECHARTS

- **Initial state**

- The object's initial state

- **States**

- A condition during the life of an object in which it satisfies some condition, performs some action, or waits for some event

- **Transitions**

- The change of state within an object

- **Events**

- An occurrence (signal from outside the system, invocation from inside the system, passage of designated period of time) that may trigger a state transition.

- **Action**

- One or more actions taken by an object in response to a state change

- **Guard**

- A boolean expression which, if true, enables an event to cause a transition

- **Final state**

- The object's final state



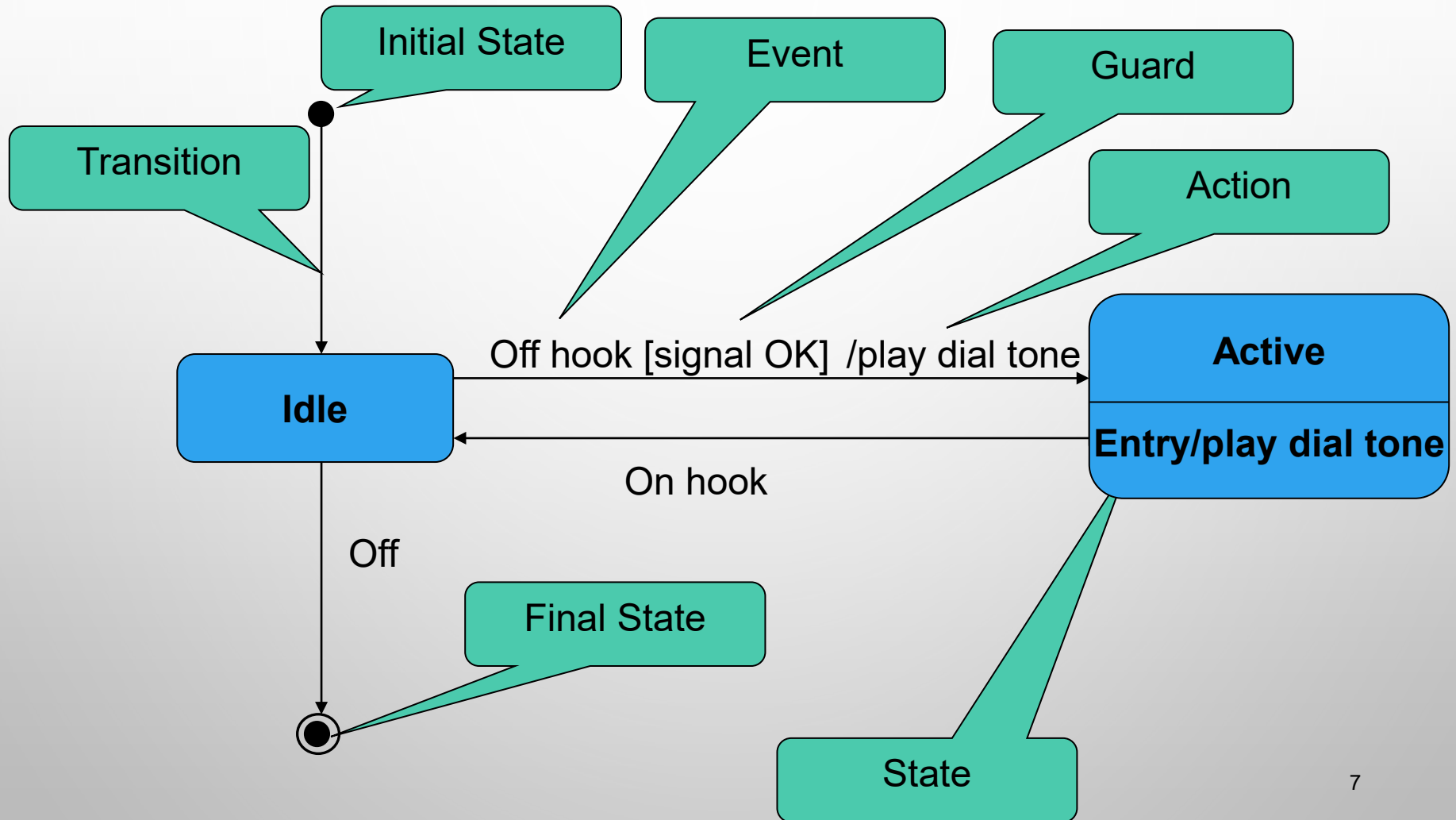
call User

/ play song

[Hungry]



AN EXAMPLE: A LAND LINE TELEPHONE CLASS



TECHNIQUES TO CREATE STATE DIAGRAMS

- Specify the class or subsystem
- Identify the distinct behaviors of the subsystem/class/object
- Identify the circumstances under which the behavior may change
- Define the types of transitions between behavior.
- Iterate

QUIZ: STATECHART DIAGRAM

- Does each state-transition diagram have one and only one initial state?
- Does each state have at least one exit transition?
- If multiple guards exist for a single event, are the guards mutually exclusive?
- Is each state and transition clearly named?
- Are all of the required states, events, guards, transitions, and actions shown on the diagram?

STATECHART DIAGRAMS GUIDELINES

- Make sure that each statechart diagram refers to only one class!
- You do not have to develop a statechart diagram for every class (unless you told so)
- All statechart diagrams should have an initial state (the state the class is when it is created)

ACTIVITY DIAGRAMS

- Activity diagrams describe the workflow behaviour of a system
- An activity diagram is a dynamic diagram that shows the activity and the event that causes the object to be in the particular state
- However, activity diagrams should not take the place of interaction diagrams and state chart diagrams. Activity diagrams do not give detail about how objects behave or how objects collaborate
- A state diagram shows the different states an object is in during the lifecycle of its existence in the system, and the transitions in the states of the objects. These transitions depict the activities causing these transitions, shown by arrows
- An activity diagram talks more about these transitions and activities causing the changes in the object states

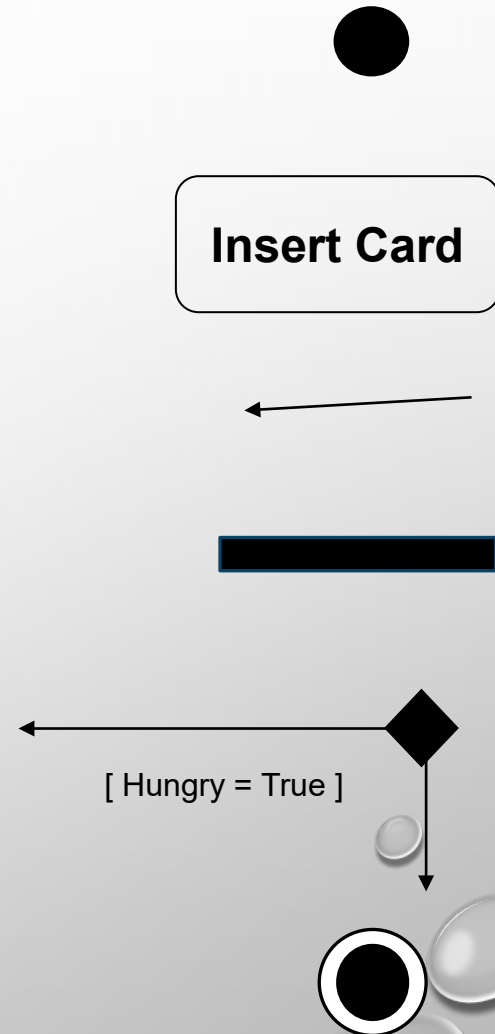
ACTIVITY DIAGRAMS

Activity diagrams are useful to

- Describe an *overall workflow* of an area of the user's activities;
- Specify how individual use cases *unfold* and may *depend* on other use cases (all transactions of an ATM follow PIN validation);
- Represent how an operation could be implemented.
- The fundamental block is an activity; a transition out of an activity normally means that the activity has been completed.

ACTIVITY DIAGRAM ELEMENTS

- Initial activity
 - The initial activity of the workflow
- Activity
 - A state of doing something. An action state is normally used to model a step in the execution of an algorithm or procedure
- Transition
 - Progression to the next activity
- Synchronisation bar (fork or join)
 - Describes concurrent activities
- Decision diamond
 - Indicates decisions and the available options
- Final activity
 - The final activity of the workflow



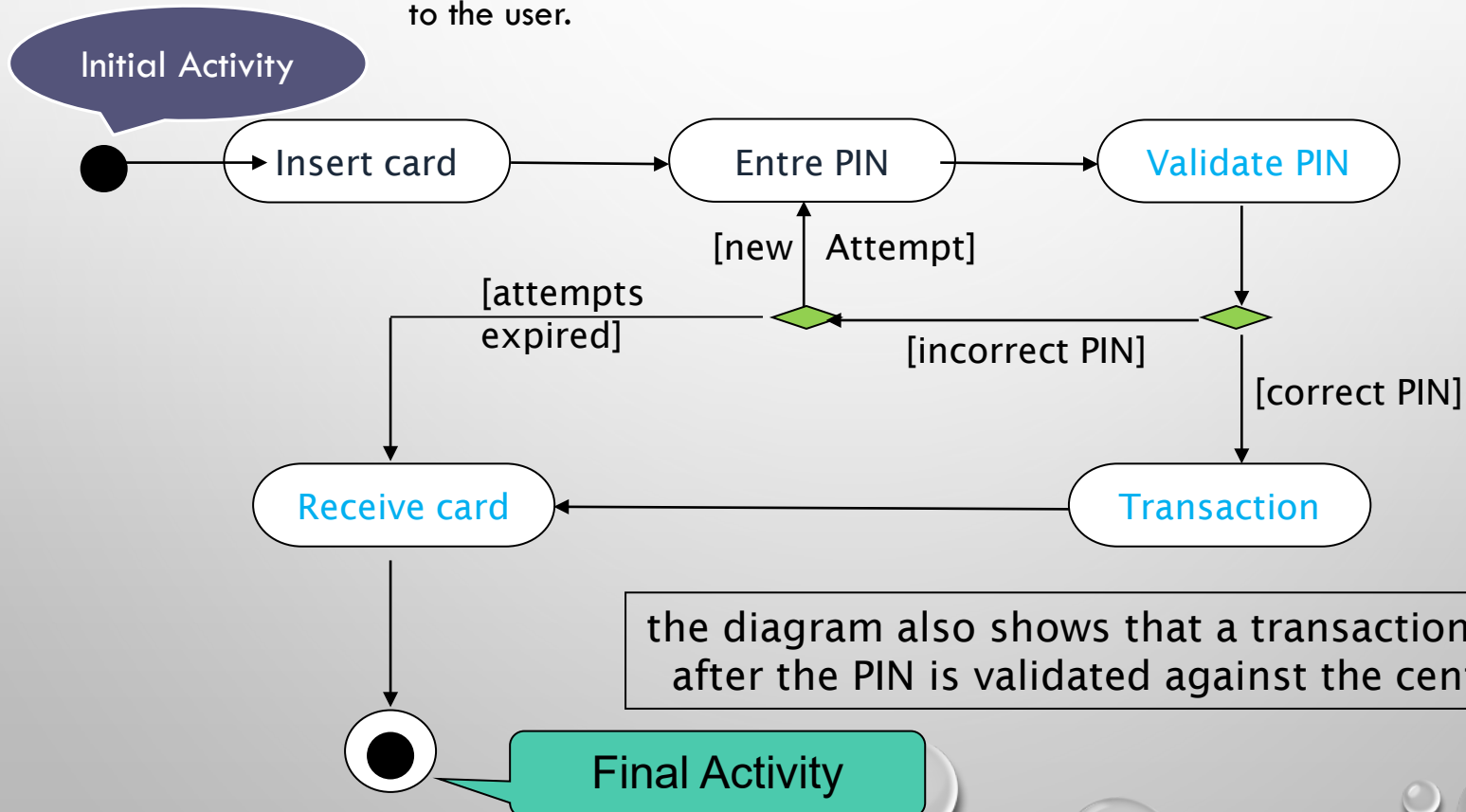
ACTIVITY DIAGRAM

- In-class activity
- Consider an **ATM** system.
- Sketch the activity of the system – from the initial activity “insert card” to the final activity “receive card”

ACTIVITY DIAGRAMS - EXAMPLE

Insert card & validate PIN:

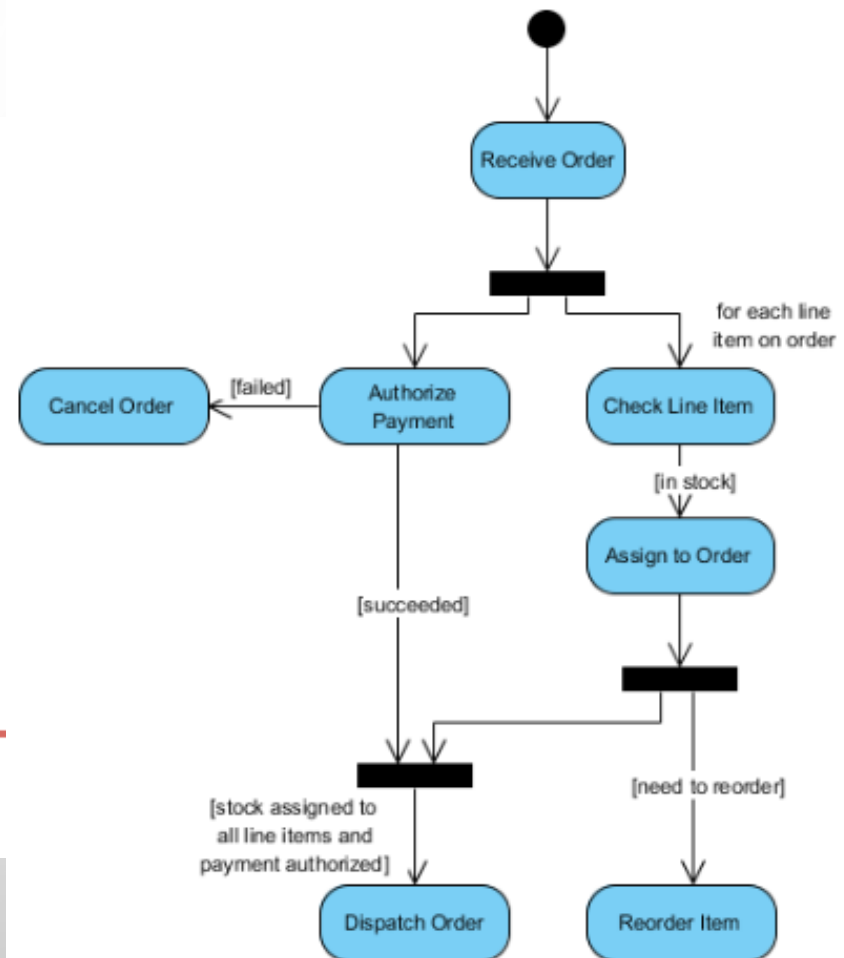
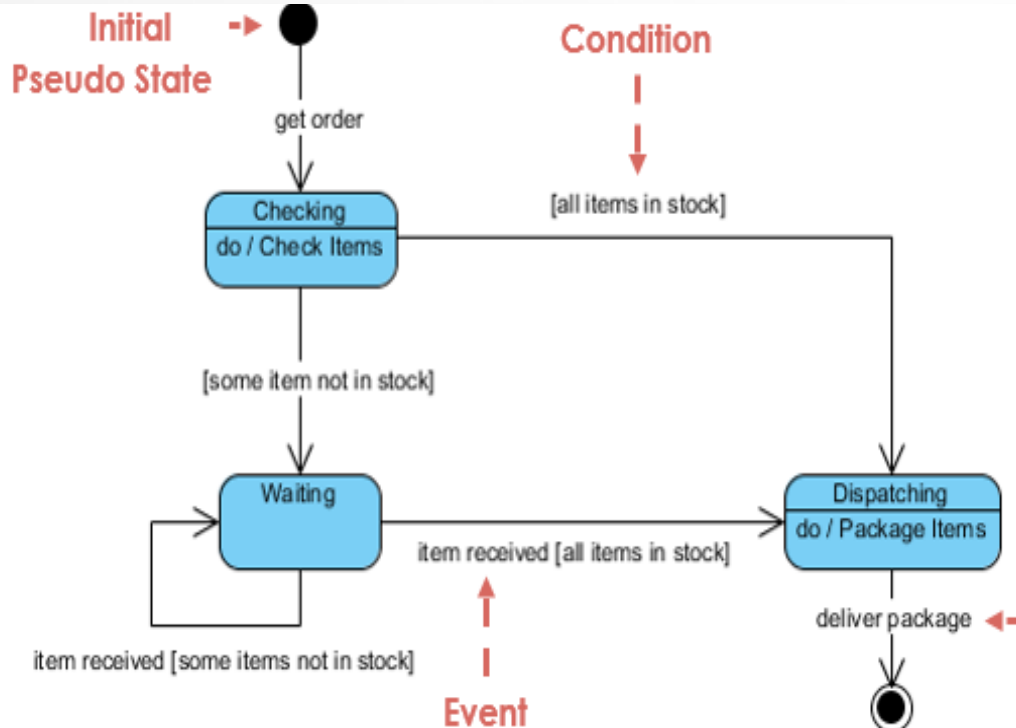
Inserts card; validates the PIN entered by the user against the central record held on the bank system. Returns an output indicating that the PIN is either correct or incorrect. If the PIN is correct then the session can continue. If the PIN is not correct then the session will terminate, and the card should be returned to the user.



USE OF ACTIVITY DIAGRAMS

- The flow of events of a use case describes the sequence of activities that together produce the value for an actor.
- An activity diagram can be used to describe the flow of events of a use case.

Statechart diagram vs Activity diagram

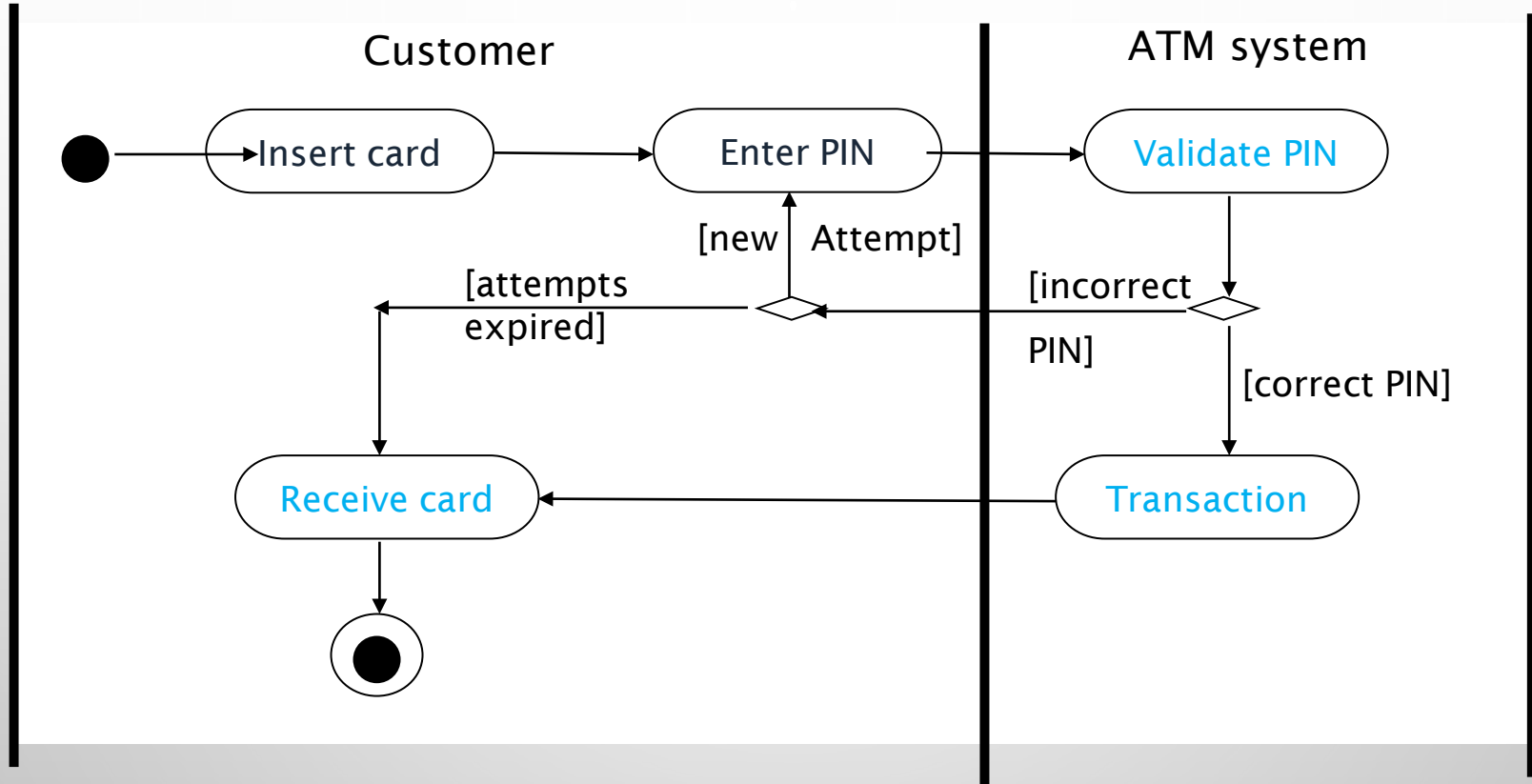


Source: (Visual Paradigm, 2023)

SWIMLANE DIAGRAM

- An activity diagram may be partitioned into swimlanes.
- Each swimlane represents a particular responsibility for some of the action states.
- The responsibilities may be allocated to a person, a part of an organisation, or a set of classes.

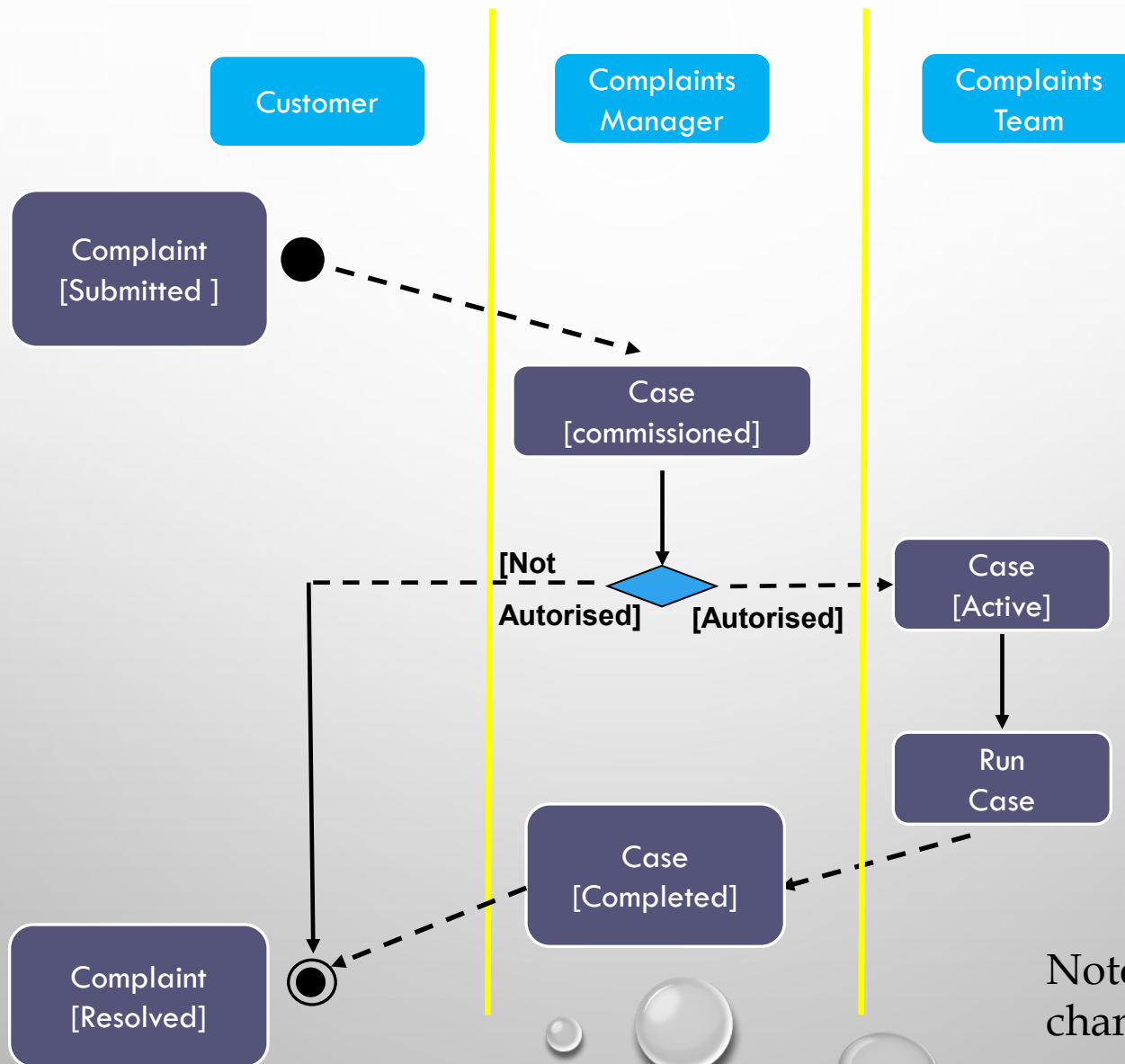
ACTIVITY DIAGRAMS - SWIMLANES



SWIMLANES - OBJECTS FLOW

- Consider a large company with the complaints department and the process of handling the complaints.
- Which objects would you envisage?
- Which activity diagram you would sketch?
- What would be the objects here
- Look at the objects flow

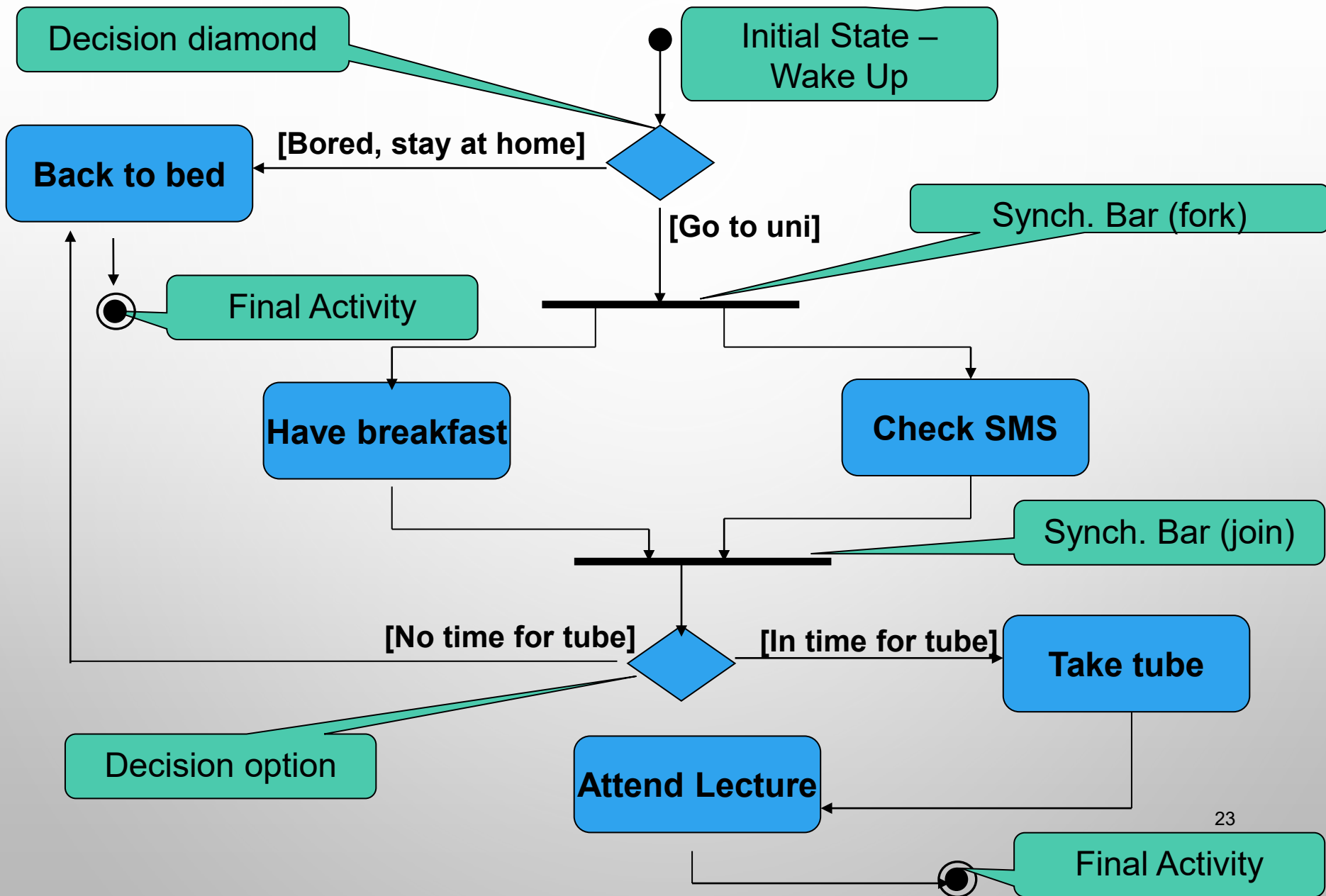
SWIMLANES - Objects flow



GUARDS AND THREADS


- Transitions show the sequence of the actions. These transitions are completion transitions.
- Guard conditions are used to represent conditional paths of execution.
- A thread of execution with a guard condition is only executed if the condition is satisfied.
- Decisions and guard conditions are used to show alternate threads.

EXAMPLE OF ACTIVITY DIAGRAM – MY LAZY DAY





OBJECT FLOWS

- Object flows are used to show how objects involved in a workflow, procedure or algorithm.
 - An object may be the output of an activity and the input activities.
 - Object flow is a kind of control flow which it replaces.
 - The object flow symbol shows an object in a particular state.
- 

QUIZ: check the code in the next two slides
and find bug in the activity diagram

SAMPLE PROGRAM OUTPUT

\$ Java menuinterface (enter key)

bicycles types

=====

1. Road

2. Mountain

3. Hybrid

4. Racing

Select option:

5 ERROR: unrecognised menu selection 5!

QUIZ: check the code in the next two slides and find bug in the activity diagram

```
Class menuinterfaceexapp {  
    // create scanner class instance  
    Private static scanner input = new scanner(system.In);  
    /* Main method */  
  
    Public static void main(string[] args) {  
        Int selector = outputmenu();  
        Processselector(selector);  
    }  
  
    /* Output menu */  
    Private static int outputmenu() {  
        System.Out.Println("bicycles types");  
        System.Out.Println("=====");  
        System.Out.Println("1. Road");  
        System.Out.Println("2. Mountain");  
        System.Out.Println("3. Hybrid");  
        System.Out.Println("4. Racing");  
        System.Out.Println("select option: ");  
        // return  
        Return(input.Nextint());  
    }  
}
```

```

/* Process selector. If unrecognised selection output error message and repeat.
*/
Private static void processselector(int selector) {
    Switch (selector) {
        Case 1:
            System.Out.Println("example bikes include kona esatto disc raleigh");
            Break;
        Case 2:
            System.Out.Println("example bikes include specialized sirrus comp disc"
            + " HOY shizuoka .004 2014 etc and C++");
            Break;
        Case 3:
            System.Out.Println("example bikes include apollo");
            Break;
        Case 4:
            System.Out.Println("example bikes include cinelli saetta radical veloce
            road bike + " miranda");
            Break;
        Default:
            System.Out.Println("error: unrecognised menu selection " +
                                selector + "!");

            selector = outputmenu();
            Processselector(selector);
    }
}
}
}

```

ACTIVITY DIAGRAM

FIND THE BUG

