TOPIC 5: STATE MACHINE DIAGRAMS

Introduction

It's necessary to have state diagrams because they help analysts, designers and developers understand the behavior of the objects in a system. Developers, in particular, have to know how objects are supposed to behave because they have to implement these behaviors in software. It's not enough to implement an object: Developers have to make that object do something.

Objectives

By the end of this topic the learner should be able to:

1. Describe state machine diagrams
2. Explain the concepts and notation used in state machine diagrams
3. Analyse and draw state machine diagrams from scenarios and text descriptions.

Learning activities

Learning Activity 5.1: Reading

Read the provided topic notes on state machine diagrams and specifically take note of the concepts and notations. Example of state machine diagrams draw from scenarios will give insight of how we draw these diagrams from scenarios and text descriptions. You have been provided with links to online resources that you can also visit to get more information on these diagrams

Learning Activity 5.2: Journal

When an order object enters the Checking state, it performs the activity "check items". After the activity is completed, the order objects transitions to the next state based on the conditions [all items available] or [an item is not available]. If an item is not available, the order is cancelled. If all items are available, then the order object is dispatched. When the order object transitions to the Dispatching state, the activity "initiate delivery" is performed. After this activity is complete, the order object transitions again to the Delivered state. Carefully read the above narrative then draw a state diagram (10 marks) *Post your article in the journal provided*

Learning Activity 5.3: Quiz

Object oriented systems are said to be systems whose the overall behavior is determined by sum of all effects of collaborating objects through message passing. Describe what happens when any one of these objects receive a message

Assessment

The scenarios in the activities 5.2 and 5.3 will be graded

Topic Resources

Online resources

https://www.youtube.com/watch?v=_6TFVzBW7oo
https://www.youtube.com/watch?v=rrqgM9Ch29Q

Topic 5 Notes

A state machine is a behavior which species the sequence of states an object visits during its life time in response to events, together with its response to those events.

When you pull a switch, a light changes its state from off to on.

Each object in a system goes through a series of states. An object state is indicated by the values of its attributes. State diagrams some of the state behavior. As the system interacts with users and with other systems, the objects that make up the system go through necessary changes to accommodate the interactions.

It's necessary to have state diagrams because they help analysts, designers and developers understand the behavior of the objects in a system. Developers, in particular, have to know how objects are supposed to behave because they have to implement these behaviors in software. It's not enough to implement an object.

NB
A state diagram shows the states of a single object
You should one state diagram for every object

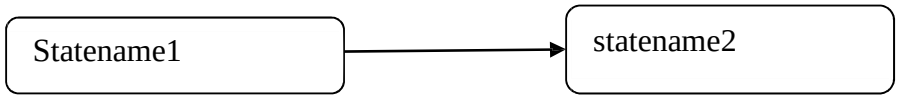
Basic concepts and notation

- **Start and end markers**

A state represents a discrete, continuous segment of time wherein the object's behavior will be stable. The object will be in a state until it is stimulated to change by an event. Each state in as state diagram is represented as a rounded rectangle with the name of the state places in it.

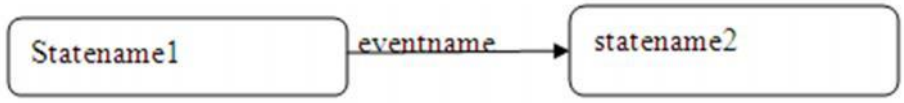
- **Transitions**

An arrow head indicates transitions between states. An object can transition from one state to another in response to various events that occur in the system



- **Event**

An event is something done to an object, such as being sent a message. State diagrams are useful for understanding how an object's reaction to a message depends on its states. When an object receives a message what it does depends on the values of its attributes.



- **Action**

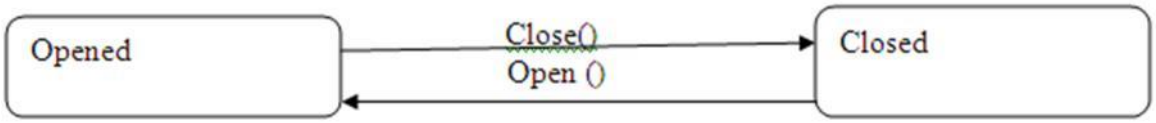
An object sending a message in response to being sent one is an example of an action being an object's reaction to an event. An action is something that the object does such as sending a message.

Examples

- **Elevator**

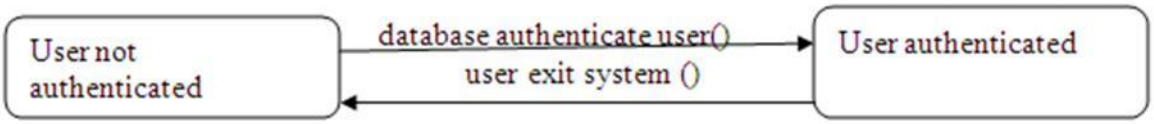
Elevator door object, assumes has attributes open, close door. It can change state from opened to

closed.



Bank system

ATM object changes from the "u**serNoAuntheticated**" to the "u**serAuthenticated**" after the database authenticates the use user after comparing the PIN the user supplies and the one in database. If the PIN is correct the ATM changes state (transitions) to the userAuthenticated and changes its userAuthenticated attribute to a value of true.ATM object transitions back to the "userNoAuntheticated" state



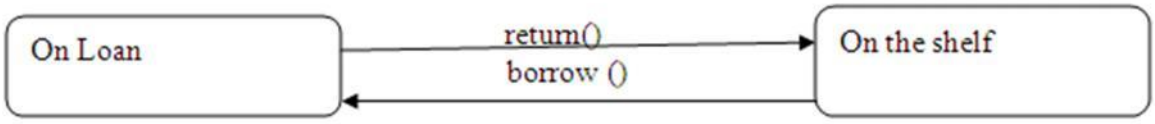
- **Library system**

One object of class copy may have a Boolean attribute onShelf, which records whether the object copy of book is currently in the library or currently on loan.

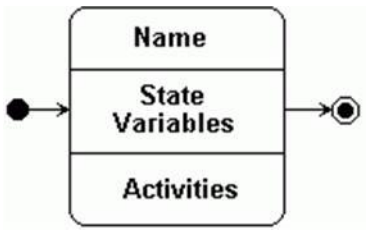
The copy object's reaction

The value of the copy's attribute onShelf is important for understanding the behavior of the object at the level of what message it seems after receiving the message itself.

We can name the two significant different states of a copy object on the shelf state and on loan state and record the message that cause it to move between the states as the events that cause transitions between the two states.



Three compartment state icon



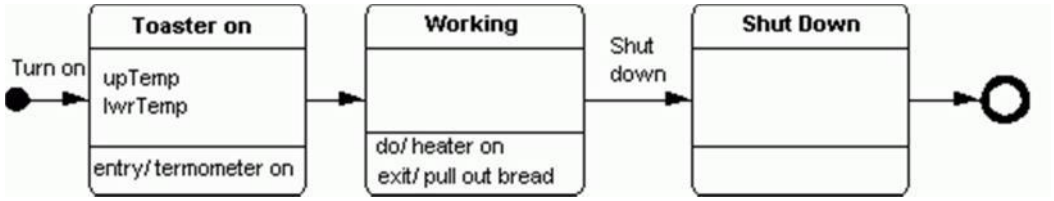
The top area holds the name of the state (this name you must supply whatever the class is divided or not), the middle area hold state variables (timers, counters, dates...), and the bottom area hold activities (***entry***-what happens when the system enters the state, ***exit***-what happens when the system leaves the state, ***do***-what happens when the system is in the state).

EXAMPLE OF A TOASTER

Suppose you're designing a toaster. You would build a plenty of UML diagrams, but here only state diagrams will be of our interest

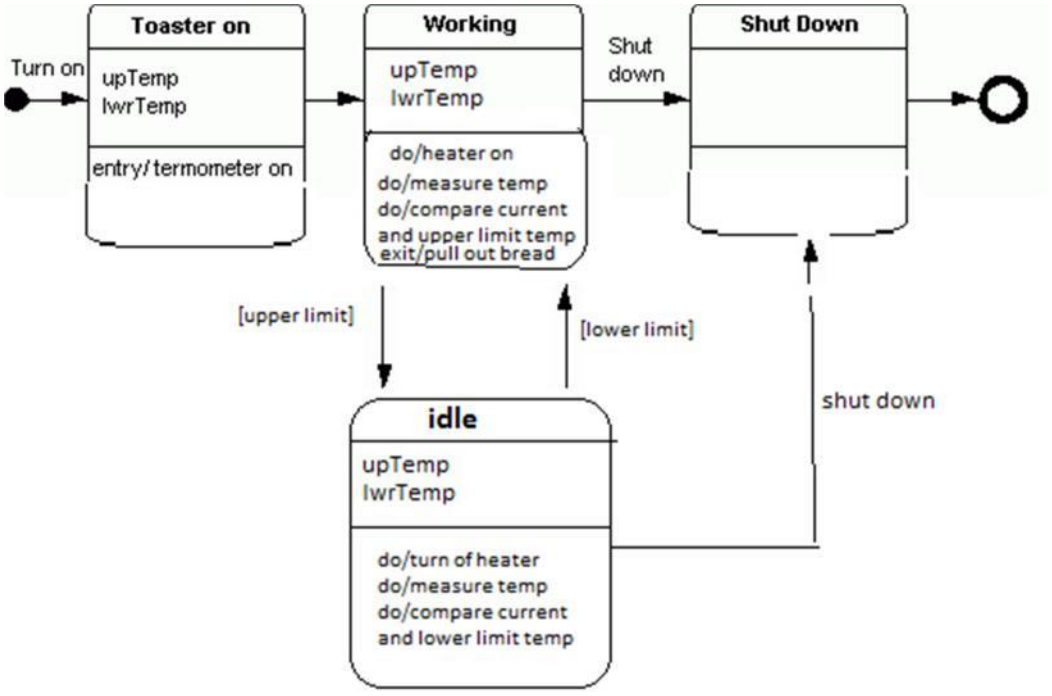
- What are the steps of making a toast?

First of all we must turn on the toaster, put in the bread and wait for several minutes to bake it. The initial state diagram is shown below:



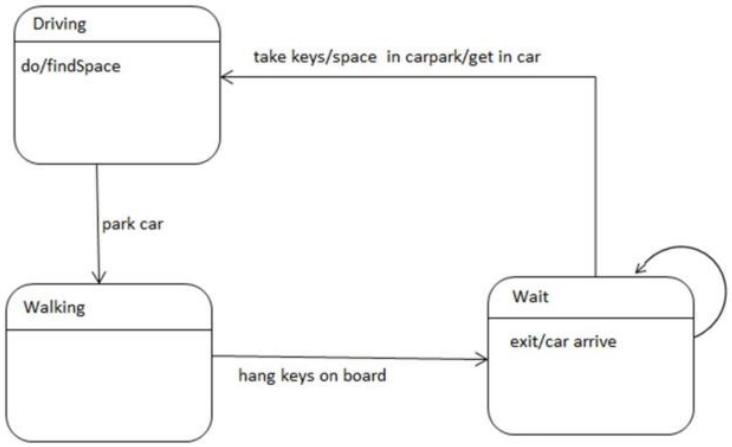
The initial state diagram for making a toast

To prevent burning out the bread, heater of the toaster must produce heat in temperature interval (upper and lower temperature limits). For this purpose thermometer measures the temperature of heater, and when the upper limit of temperature is reached then heater must go into idle state. This state resists until heater's temperature decreases to lower limit, and then working state is again aimed. With this new state, extended state diagram will be:



Revision Questions

The following state diagram models the behavior of a drop-off car park attendant. The attendant's wait at the drop-ff ate where cars arrive and their owners hand over the keys ready for parking. If the car park is not full, the attendant drives the car to an empty parking space then walks back to the kiosk where he places the keys on a board used by the pick-up car park attendants, The pick-up park attendant waits at the pick-up gate until a customer returns asking for their car. The pick-up attendant checks the keys are the on the board, goes walking I the car park to search for the car, and the drives it back to the kiosk and hands the keys to the customer. The pick-up attendant then waits for the next customer. Draw a state diagram to model the behaviour of the pick-up car attendant. Label each element of your diagram explaining its purpose



TOPIC 6: ACTIVITY DIAGRAMS

Introduction

Activity diagram gives a simplified look of what happens during an operation or a process. Activity diagrams describe the workflow behavior of a system. Activity diagrams should be used in conjunction with other modeling techniques such as interaction diagrams and state diagrams. Activity diagrams analyze a use case by describing what actions need to take place and when they should occur.

Objectives

By the end of this topic you should be able to:

1. Describe activity diagram
2. Explain the concepts and notation used in activity diagram
3. Analysis scenarios and text descriptions and draw activity diagrams from them.

Learning activities

Learning Activity 6.1: Reading

Read the provided topic notes on activity diagrams and specifically take note of the concepts and notations. Example of activity diagrams draw from scenarios will give insight of how we draw these diagrams from scenarios and text descriptions. You have been provided with links to online resources that you can also visit to get more information on these diagrams

Learning Activity 6.2: Journal

When Jane gets to her house after work, she performs various activities. She gathers her laundry. If there is enough detergents in the house, she washes her clothes using her washing machine. If there isn't enough detergent in the house, she takes her laundry to a nearby laundry service (who do instant dry cleaning), gets her now washed laundry from the laundry service, buys detergent, and goes back to the house. She then fixes supper. She washes the utensils even as she cleans the house. She then has her supper. Draw activity diagram from the above text descript (10 marks)

Learning Activity 6.3: Discussion

Consider the following flight check-in subsystem requirements. In order to successfully check-in passengers, the attendant must weigh the passengers' luggage and assign them a seat. There are two ways to assign a seat: assigning a window seat and assigning an aisle seat, but only one need be completed in the process of assigning the passenger a seat. Create an activity diagram for the above problem (7 marks)

Assessment

The diagrams drawn from the text descriptions in the activity 6.2 and 6.3 will be graded.

Topic Resources

Online resources

https://www.youtube.com/watch?v=_tCedK0CfMk

Topic 6 Notes

Activity diagram is much like the flowcharts. A flowchart shows sequence of steps, processes, decisions points and branches. Activity diagrams show steps (activities) as well as decision points and branches. It is useful to show what happens in business process or an operation. Activity diagrams provide illustrations of what happens in a work flow, what activities can be done in parallel and whether there are alternative paths through the work flow. Activity diagram gives a simplified look of what happens during an operation or a process. Activity diagrams describe the workflow behavior of a system. Activity diagrams should be used in conjunction with other modeling techniques such as interaction diagrams and state diagrams. Activity diagrams analyze a use case by describing what actions need to take place and when they should occur. A complete workflow description will have a basic flow, and one or several alternative flows. This workflow has a structure that we can define textually, using informal if, if-then-else, or do -until statements of various kinds. Activity diagrams show be read from top to bottom.

Basic Activity Diagram Notation

The activity diagram notation has some elements that are necessary for you to understand if you want to be "conversant" about activity diagrams.

Start markers marks the beginning
Activity states/activity states, which represent the performance of a step within the workflow. It is shown as named box with rounded corners.
Transitions that show what activity state follows after another. This type of transition is sometimes referred to as a **completion transition**, since it differs from a transition in that it does not require an explicit trigger event, it is triggered by the completion of the activity the activity state represents.
Transitions are not labeled with events or actions because they fire when the previous activity completes not because of an outside event. However transitions can be labeled with guard conditions, if the next activity depends on the situation

Decisions for which a set of ***guard conditions*** are defined. These guard conditions control which transition of a set of alternative transitions that follows once the activity has been completed.
You may also use the decision icon to show where the threads merge again. Decisions and guard conditions allow you to show *alternative threads* in the workflow of a business use case.

Decision diamond

Used to show decisions as alternatives to guards on separate edges (*alternative threads)* leaving the same state. Decisions also show where the threads merge again.

Branch

The branch describes what activity will take place based on a set of conditions.

Merge

All braches at some point merge to indicate the end of the conditional behavior which was started by the branch.

Synchronization bar

It is thick horizontal bar describing the coordination of activities. It provides a way to express things such as waiting for all subtasks to finish before proceeding (join). Synchronization bars, which you can use to show parallel subflows. Synchronization bars allow you to show concurrent threads in the workflow of a business use case.

Synchronization bar (fork)

A fork is used when multiple activities are occurring at the same time.

Synchronization bar (join)

After the merge all of the parallel activities must be combined by a join before transitioning into the final activity state

Partitions/swimlanes

The contents of an activity diagram may be organized into partitions using solid vertical lines. Partitions are used to represent an organizational unit of some king. Partitions add the dimensions of visualizing roles **(figure 3)** . Each partition or swimlane shows the name of the role at the top and represents the activities of each role. Swimlanes contain several groups of related activities and transitions can place fro one swimlane to another

Conditional Threads

You describe which one of a set of alternative threads to follow by defining guard conditions for each thread.

Guard conditions can also be used to show that one of a set of concurrent threads is conditional. For example, in the individual check in example from figure 1, the passenger checking in might be a frequent-flyer member. In that case, you need to award the passenger frequent flyer miles.

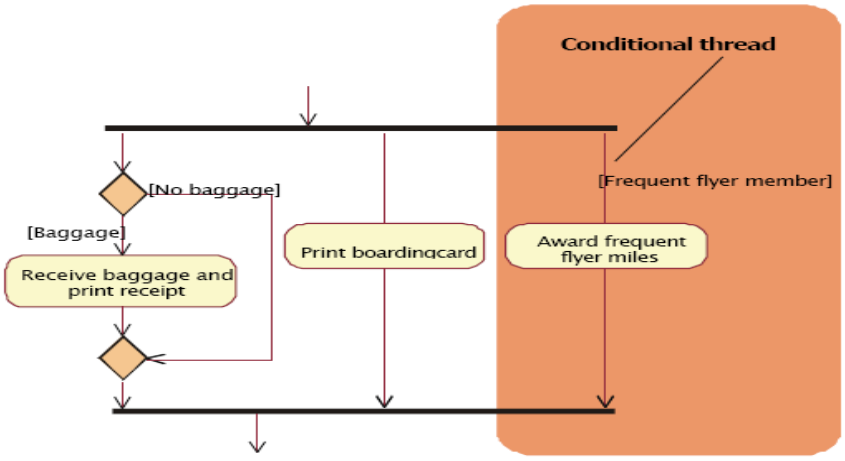
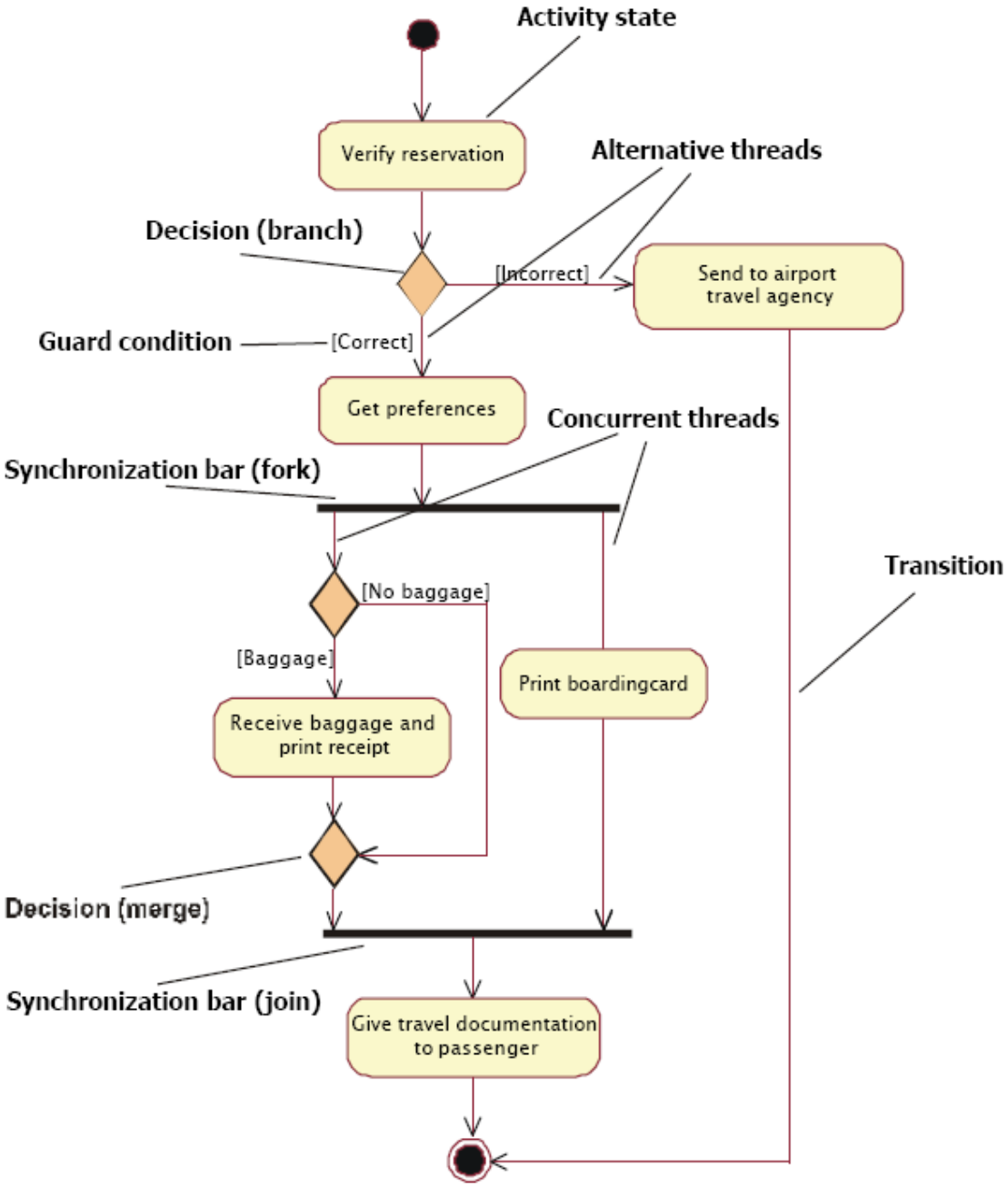


Figure 2. Awarding frequent flyer miles is a conditional thread in the Individual Check-In workflow.


Activity state
Verify reservation
Alternative threads
Decision (branch)
[Incorrect]
Send to airport travel agency
Guard condition
[Correct]
Get preferences
Concurrent threads
Synchronization bar (fork)
Transition
[No baggage]
[Baggage]
Print boardingcard
Receive baggage and print receipt
Decision (merge)
Synchronization bar (join)
Give travel documentation to passenger

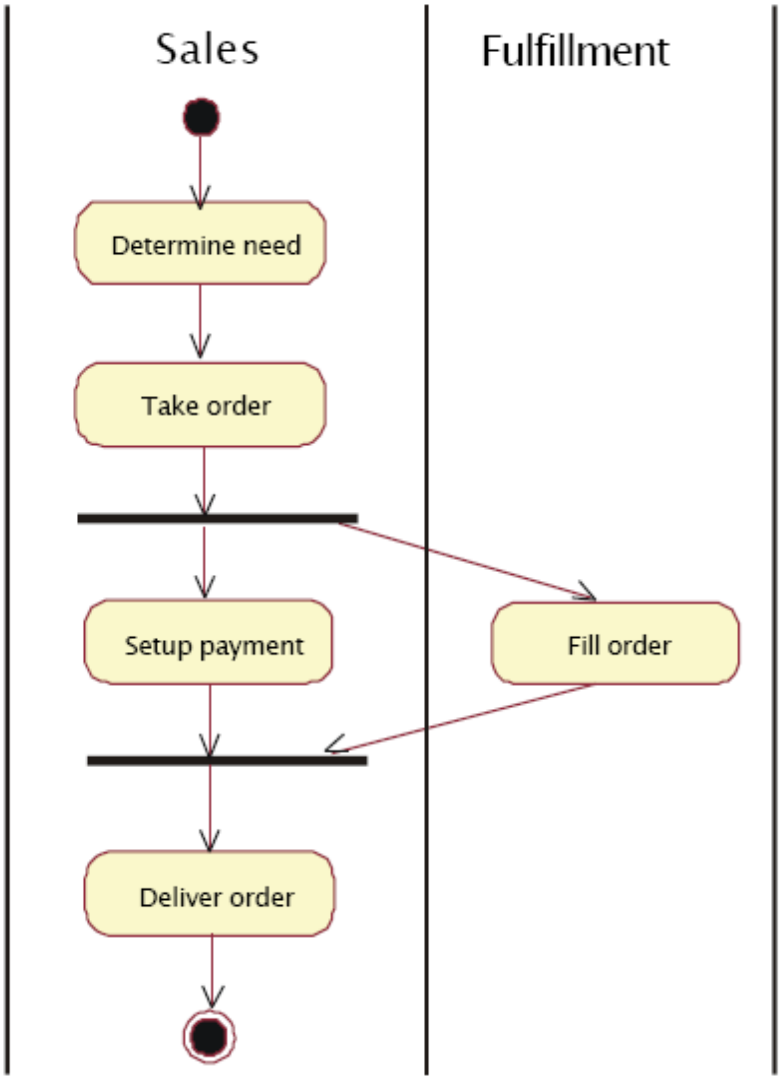
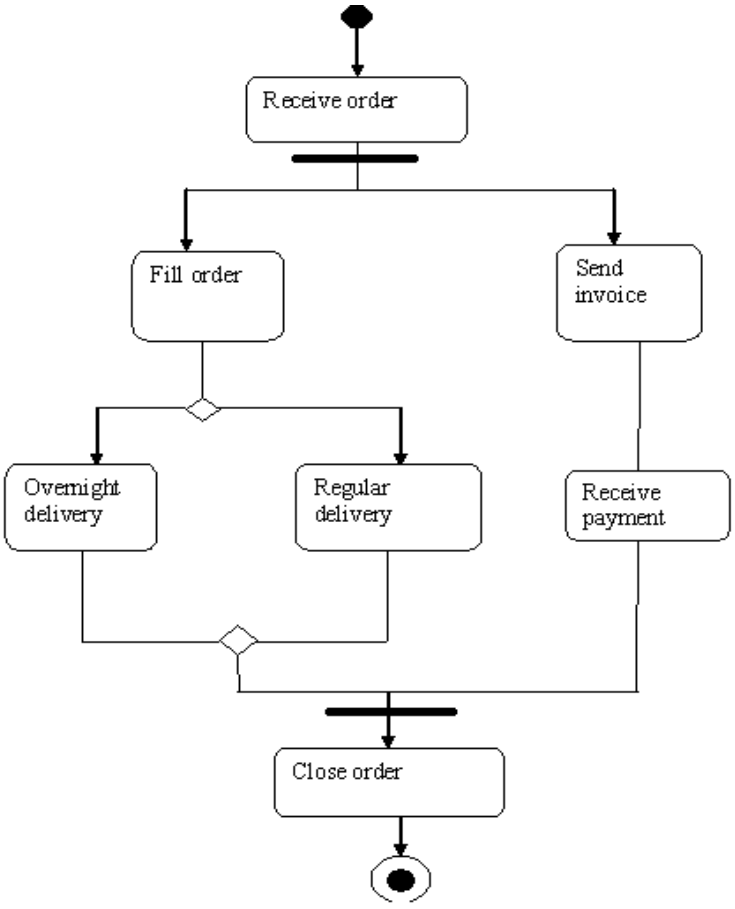


Figure 3. An activity diagram illustrating the workflow of a business use case that represents a (generic) sales process. In this example, the partitions represent departments in the organization.

Examples

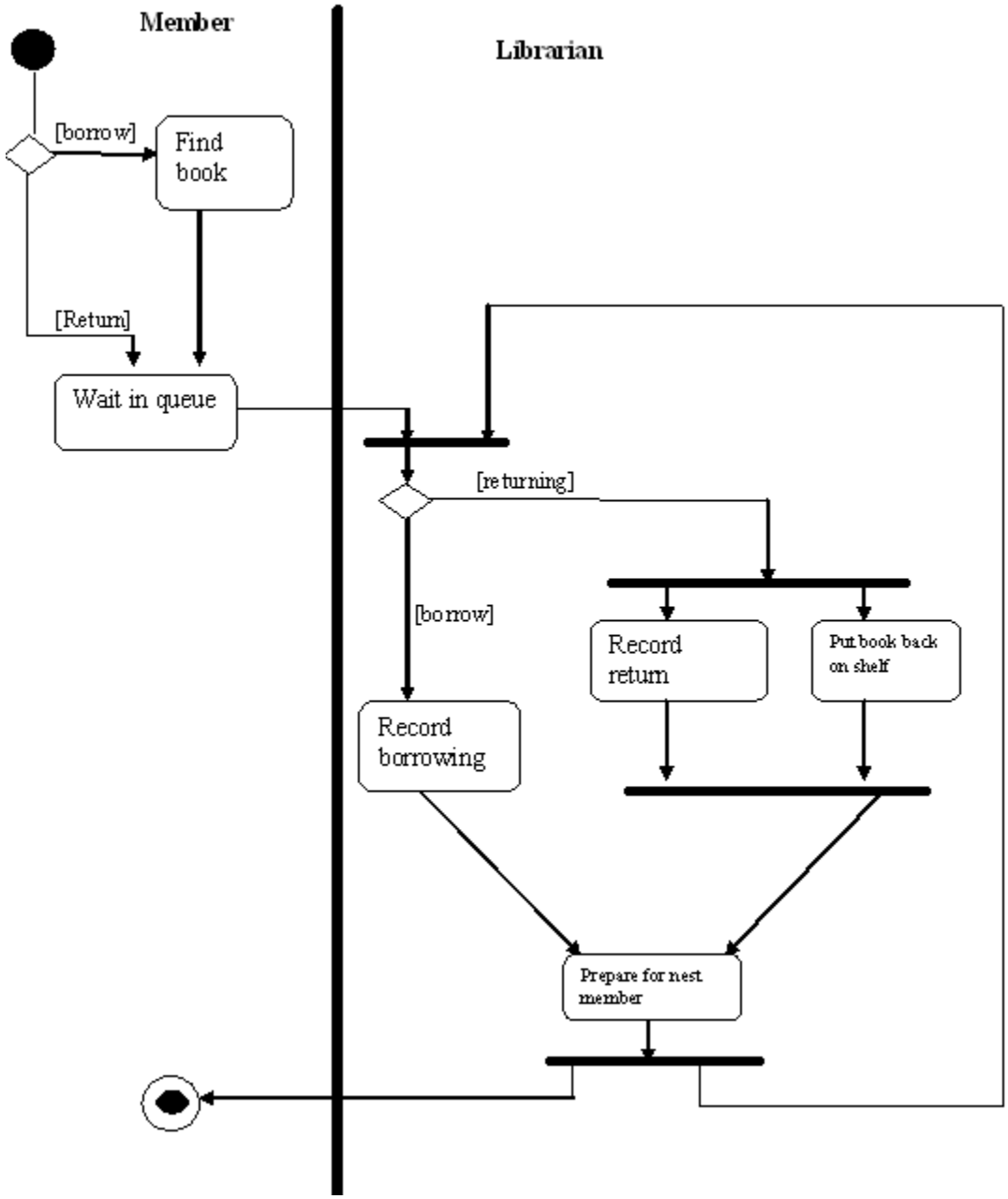
Order processing

Once order is received the activities split into two parallel sets of activities, one side fills and sends the order while the other handles the billing. On the filling order the method of delivery is decided conditionally, depending on the condition either overnight delivery or regular delivery activity is performed.



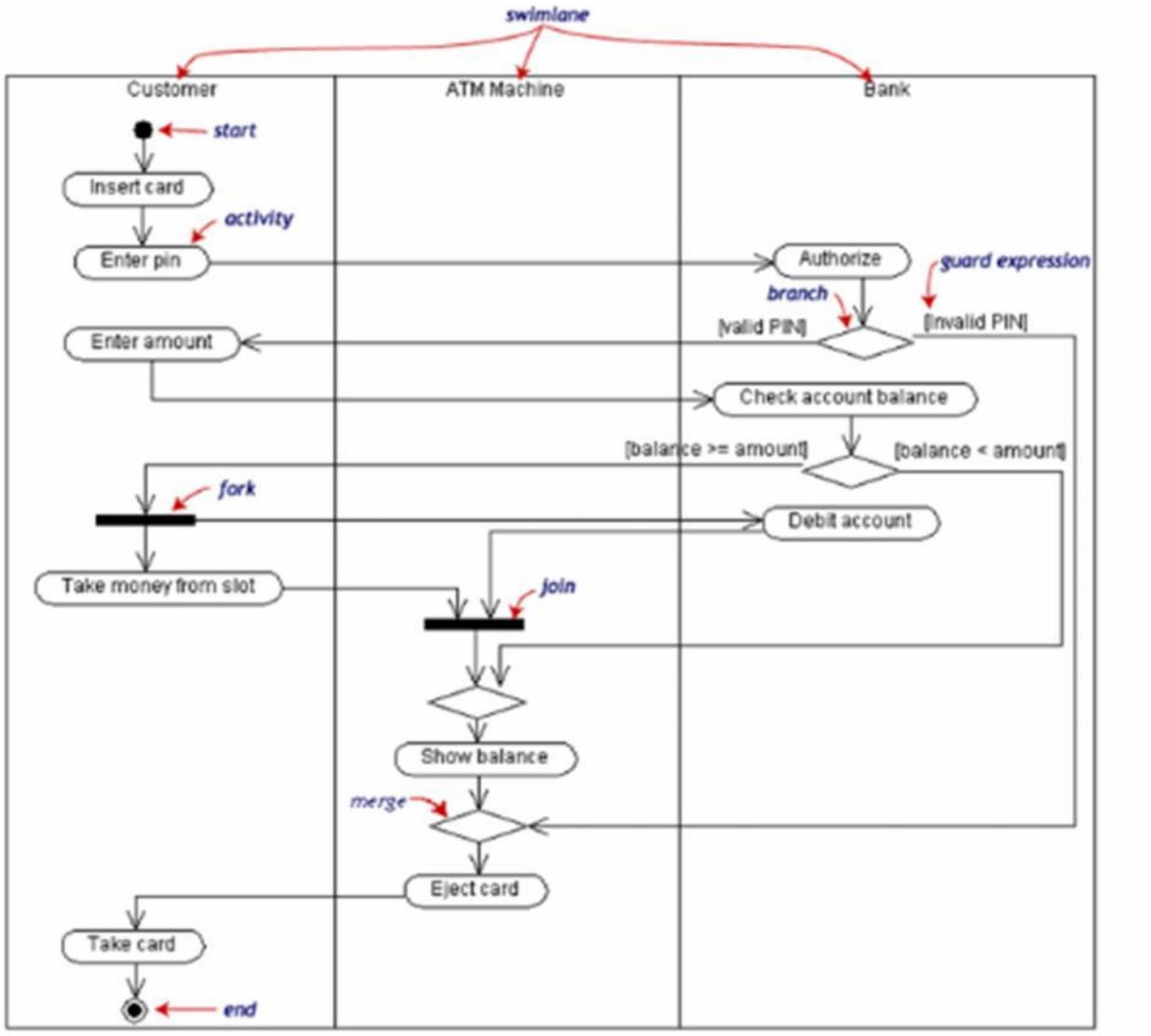
The library contains books and journals. It may have several copies of a given book. Some of the books are for short term loans only. All other books may be borrowed by any library member for three weeks. Members of the library can normally borrow up to six items at a time, but members of staff may borrow up to 12 items at any one time. Only members of staff may borrow journals.

Model the problem using an activity diagram (10 marks)



Withdraw money from a bank account through an ATM.

The three involved classes (people, etc.) of the activity are **Customer**, **ATM**, and **Bank**. The process begins at the black start circle at the top and ends at the concentric white/black stop circles at the bottom. The activities are rounded rectangles.



Revision Questions

Buttons in elevators and on the floors control the movement of 8 elevators in a building in building with 20 floors. Buttons illuminate when pressed to request an elevator to stop at a specific floor. The illumination is cancelled when the request has been satisfied. When an elevator has no requests it remains at its current floor with its doors closed. Describe the messages being exchanged by these objects