Object Oriented Analysis and Design with Java

UE19CS353

Prof. Sudeepa Roy Dey and Prof. Mahitha G
Department of Computer Science and Engineering

Acknowledgements: Significant portions of the information in the slide sets presented through the course in the class, are extracted from the prescribed text books, information from the Internet. Since these are only intended for presentation for teaching within PESU, there was no explicit permission solicited. We would like to sincerely thank and acknowledge that the credit/rights remain with the original authors/creators only

PES UNIVERSITY ONLINE

UE19CS353: Object Oriented Analysis and Design with Java

Case studies on Activity Diagram and State Modeling

Recap your concepts

Activity Diagram vs. Statechart Diagram

Statechart Diagram for Incident

Focus on the set of attributes of a single abstraction (object, system)

Activity Diagram for Incident

(Focus on dataflow in a system)

Object Oriented Analysis and Design with Java

Case study 1: Managing Course Information Activity diagram

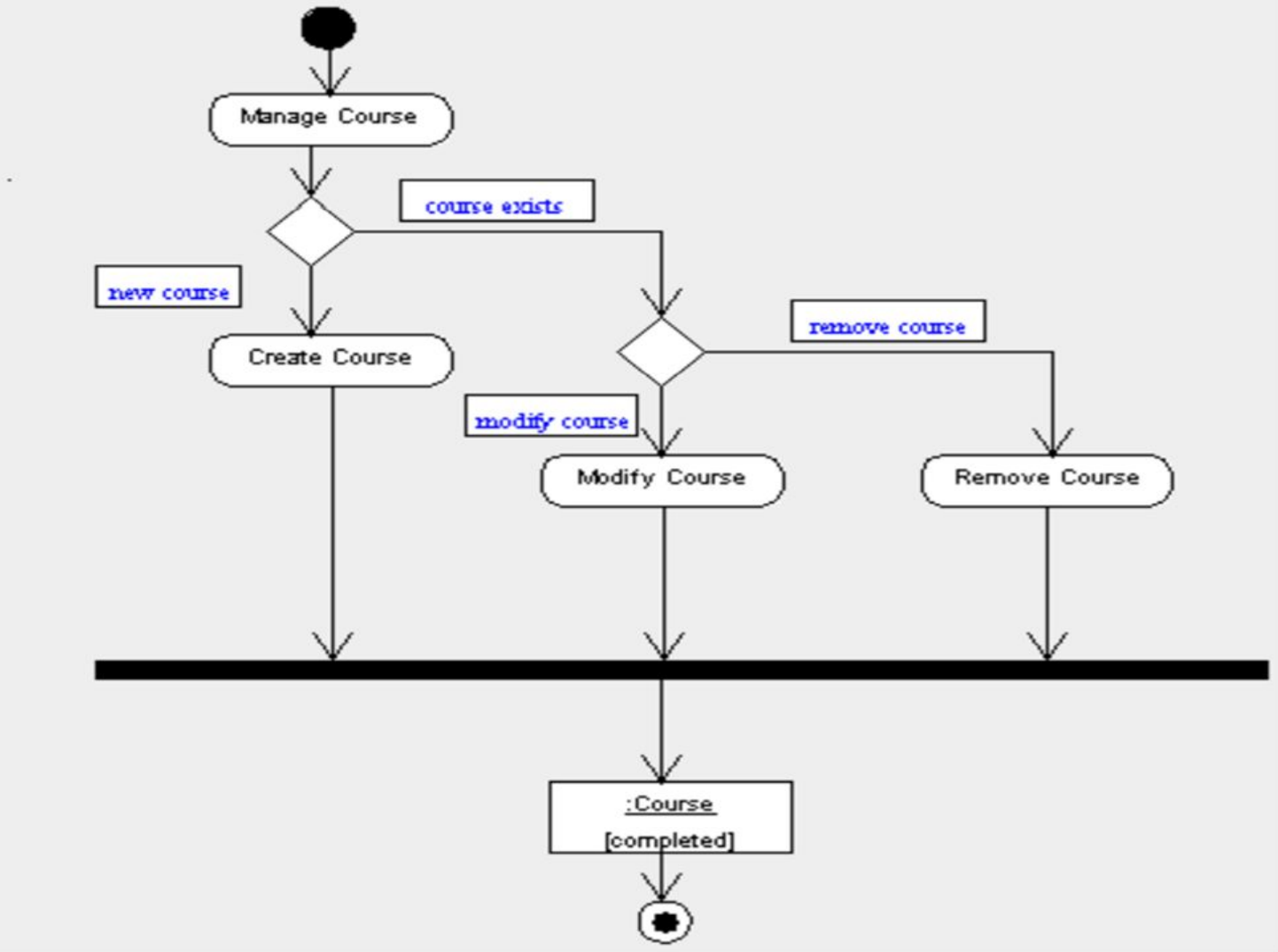
The course administrator is responsible for managing course information in the Course Management System by following activities:

- Check if course exists
- If course is new, proceed to the "Create Course" step
- If course exists, check what operation is desired—whether to modify the course or remove the course
- If the modify course operation is selected by the course administrator, the "Modify Course" activity is performed
- If the remove course operation is selected by the course administrator, the "Remove Course" activity is performed. In the first step in this Activity diagram, the system determines whether the course that is to be managed is a new course or an existing course.

For managing a new course, a separate activity, "Create Course," is performed. On the other hand, if a course exists, the course administrator can perform two different activities—modify an existing course or remove an existing course. Hence, the system checks the type of operation desired based on which two separate activities can be performed—"Modify Course" or "Remove Course".

Object Oriented Analysis and Design with Java

Solution– Activity Diagram of Managing Course



Explanation– Activity Diagram

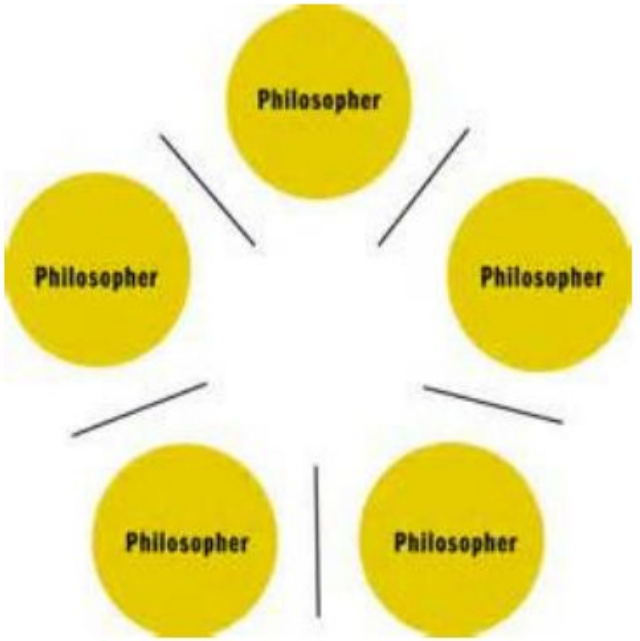
1. In the first step in this Activity diagram, the system determines whether the course that is to be managed is a new course or an existing course.
2. For managing a new course, a separate activity, "Create Course," is performed.
3. On the other hand, if a course exists, the course administrator can perform two different activities—modify an existing course or remove an existing course.
4. Hence, the system checks the type of operation desired based on which two separate activities can be performed—"Modify Course" or "Remove Course"

Object Oriented Analysis and Design with Java

Case study 2– Dining Philosophers Problem

Problem: Five silent philosophers sit at a round table with bowls of noodles. A chopstick is placed between each pair of adjacent philosophers. Each philosopher must alternately think and eat. However, a philosopher can only eat noodles when he has both “left” and “right” chopsticks. Each chopstick can be held by only one philosopher and so a philosopher can use the chopstick only if it is not being used by another philosopher. After he finishes eating, he needs to put down both chopsticks so they become available to others. A philosopher can take the chopstick on his right or the one on his left as they become available, but cannot start eating before getting both of them, Nor can he take a chopstick that is not immediately on his left or right. The problem is how to design a discipline of behaviour such that no philosopher will starve

Figure 1 The Dining Philosophers



Interesting Problem – Dining Philosophers Problem

- Identify the activities and transitions of a philosopher.
- Each chopstick is assigned a “partial order” value (“0” to “4”) (with no duplication) so each philosopher has a “lower ordered” chopstick on one side and a “higher ordered” chopstick on the other side.
- Create an **activity diagram for one philosopher** by imposing the rule that the lower ordered chopstick must be picked up first.

The philosophers cannot communicate with each other*.

Object Oriented Analysis and Design with Java

Activity Diagram- Case study 2 :solution

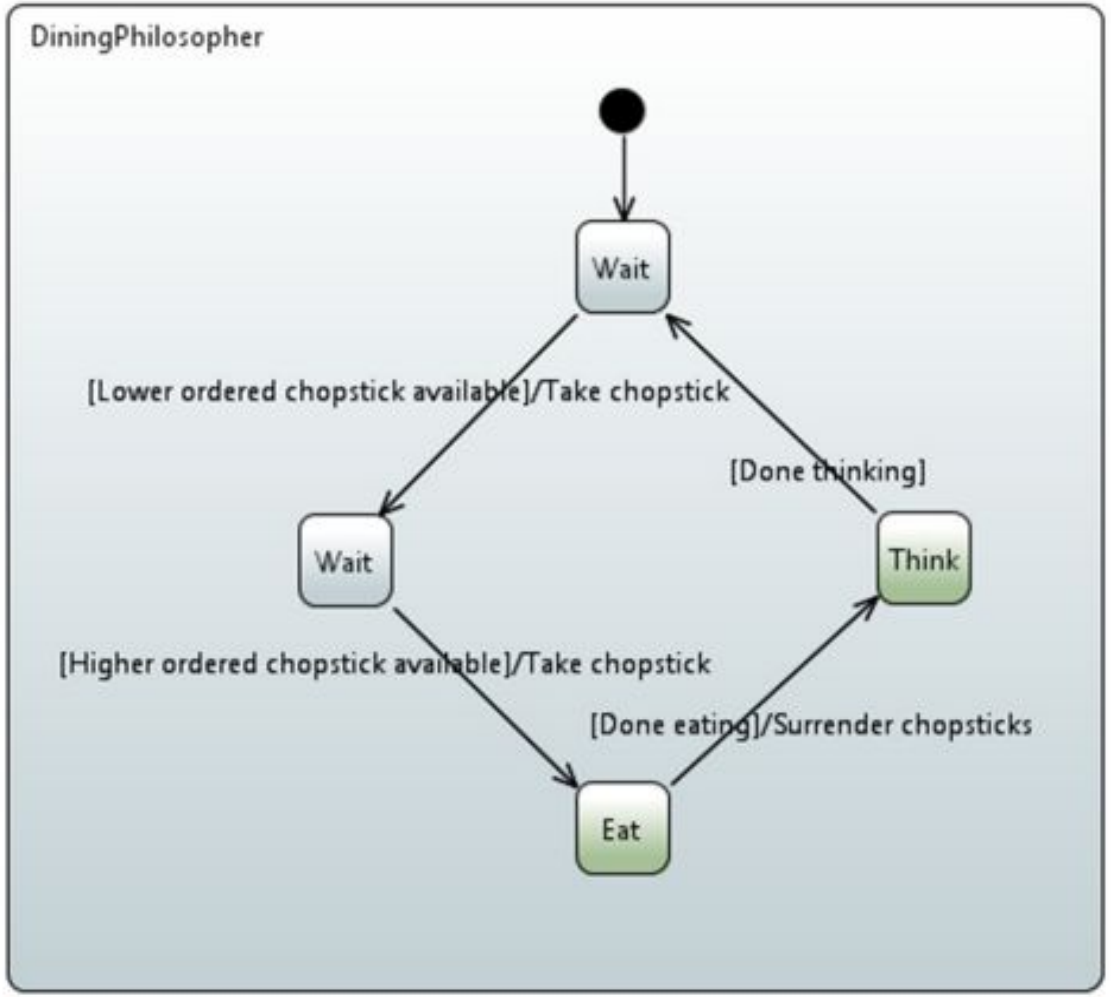


Image source:http://www.ecoa.technology/_static/tutorials/developing/DiningPhilosophers.pdf

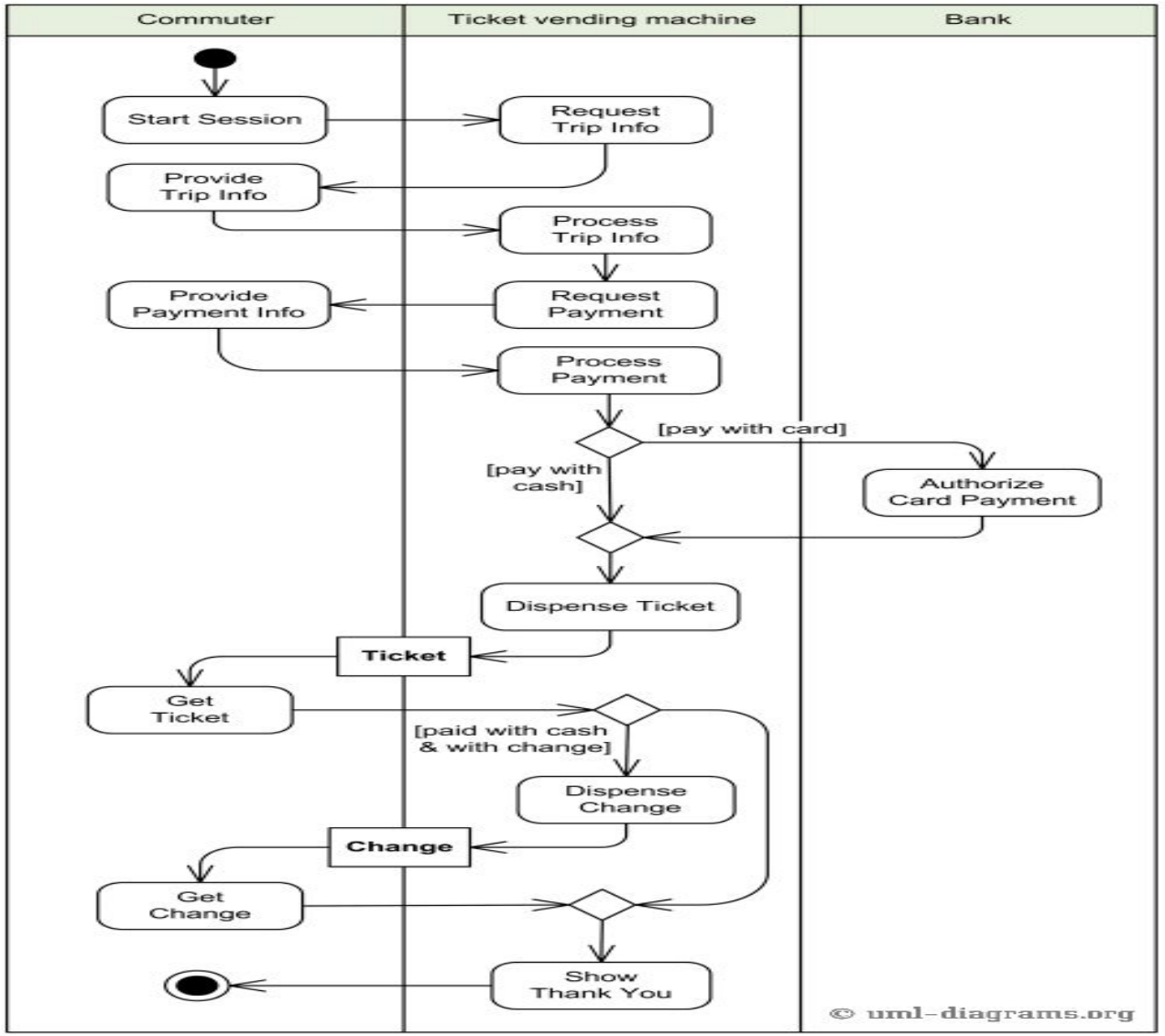
Object Oriented Analysis and Design with Java

Activity Diagram with swim lanes: Case study 3 on Purchase ticket

- Activity is started by Commuter actor who needs to buy a ticket. Ticket vending machine will request trip information from Commuter. This information will include number and type of tickets, e.g. whether it is a monthly pass, one way or round ticket, route number, destination or zone number, etc.

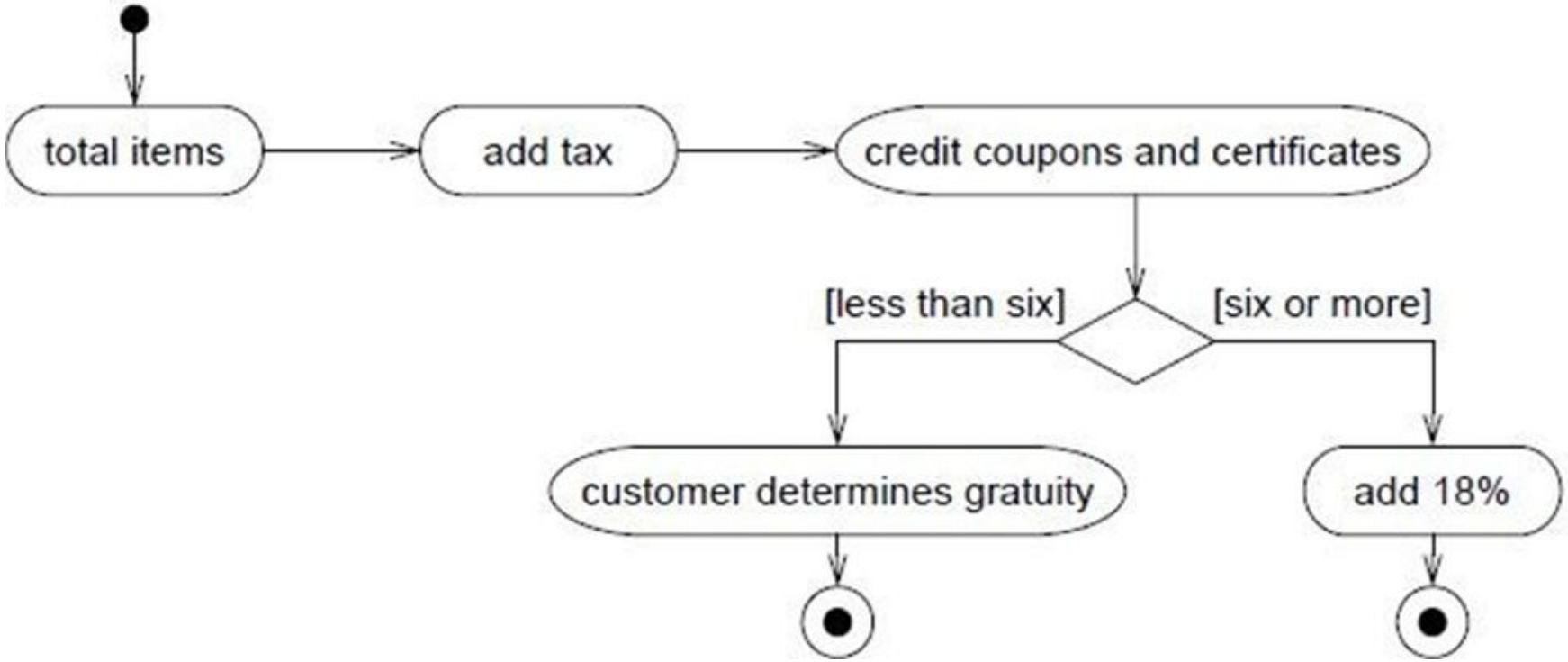
- Based on the provided trip info ticket vending machine will calculate payment due and request payment options. Those options include payment by cash, or by credit or debit card. If payment by card was selected by Commuter, another actor, Bank will participate in the activity by authorizing the payment.

Solution – Activity Diagram of Ticket Vending machine



Additional Problems

Prepare an activity diagram for computing a restaurant bill. There should be a charge for each delivered item. The total amount should be subject to tax and a service charge of 18% for groups of six of more. For smaller groups, there should be a blank entry for a gratuity according to the customer's discretion. Any coupons or gift certificates submitted by the customer should be subtracted.



Case studies

on

State Modeling

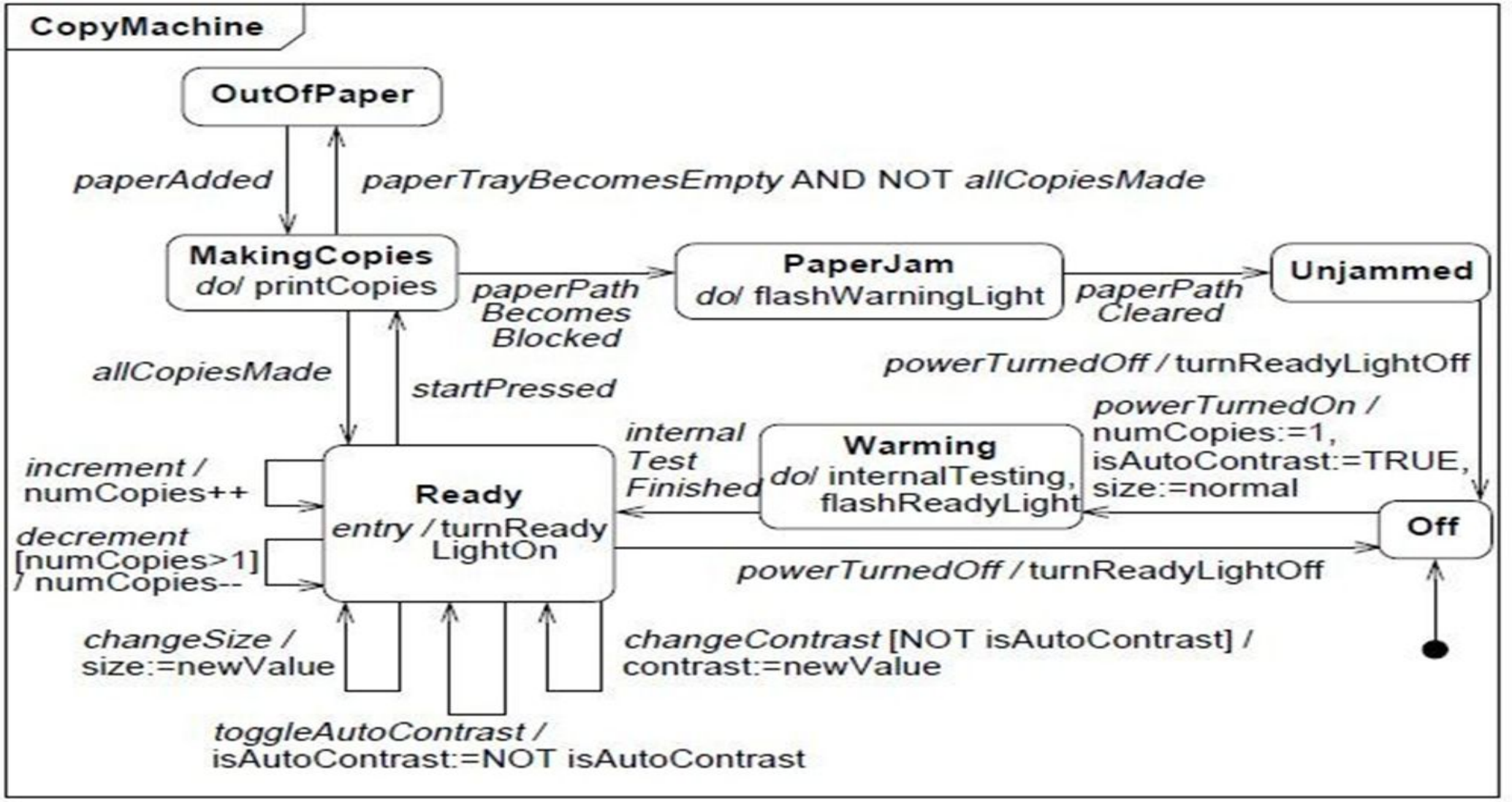
Object Oriented Analysis and Design with Java

To exhibit skills on modeling concepts learnt – State Diagram

Identify states and events for a Photocopier (Zerox) machine from the description given below and draw the state diagram for the same. Initially the machine is off. When the operator switches on the machine, it first warms up during which it performs some internal tests. Once the tests are over, machine is ready for making copies. When operator loads a page to be photocopied and press 'start' button, machine starts making copies according to the number of copies selected. While machine is making copies, machine may go out of paper. Once operator loads sufficient pages, it can start making copies again. During the photocopy process, if paper jam occurs in the machine, operator may need to clean the path by removing the jammed paper to make the machine ready. You may add more events as per question as well.

Object Oriented Analysis and Design with Java

Solution



Object Oriented Analysis and Design with Java

To exhibit skills on modeling concepts learnt – State Diagram

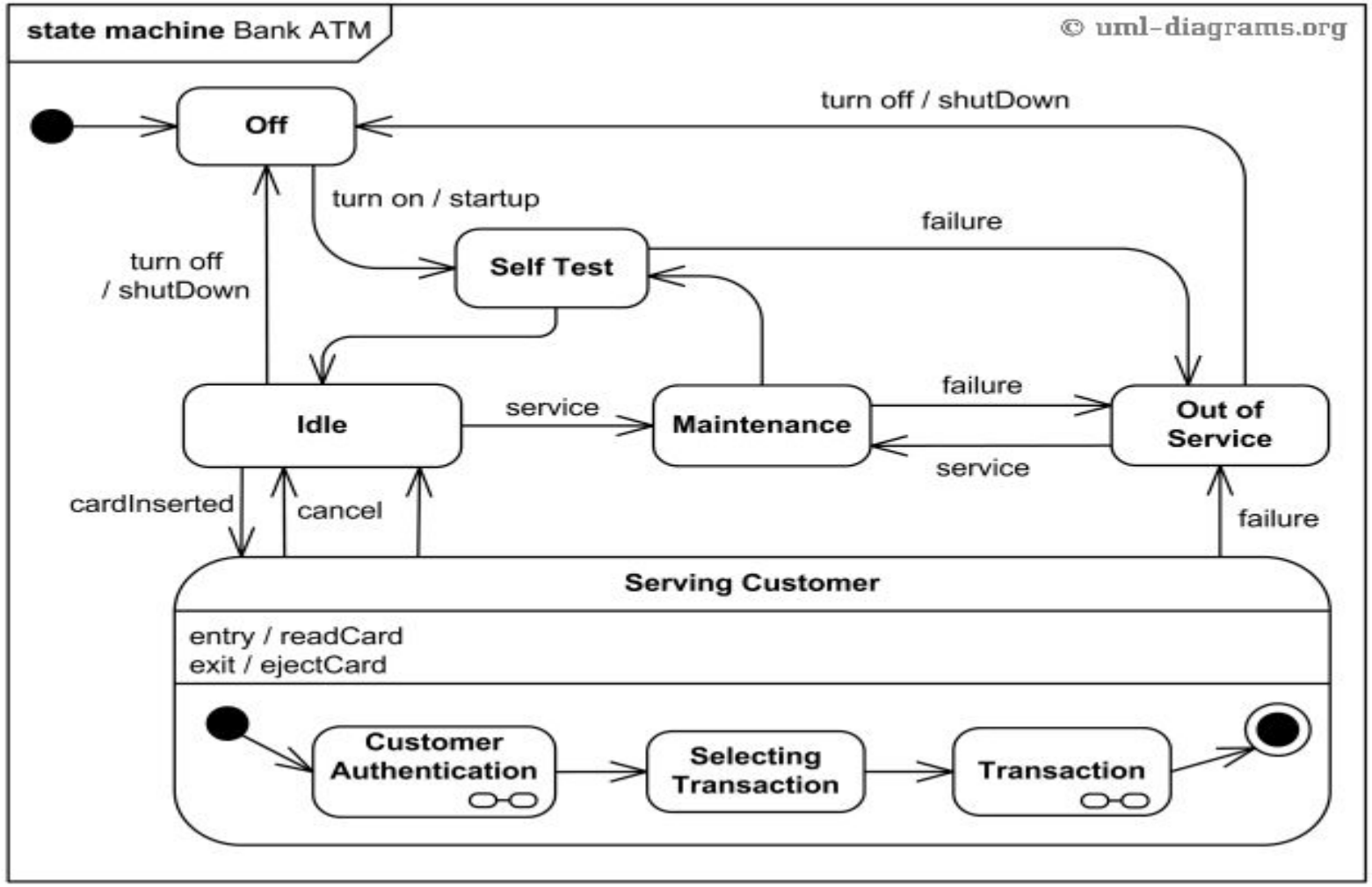
Case study on Bank Automated Teller Machine (ATM)

- ATM is initially turned off. After the power is turned on, ATM performs startup action and enters Self Test state. If the test fails, ATM goes into Out of Service state, otherwise there is triggerless transition to the Idle state. In this state ATM waits for customer interaction.

- The ATM state changes from Idle to Serving Customer when the customer inserts banking or credit card in the ATM's card reader. On entering the Serving Customer state, the entry action readCard is performed. Note, that transition from Serving Customer state back to the Idle state could be triggered by cancel event as the customer could cancel transaction at any time.

Object Oriented Analysis and Design with Java

Solution

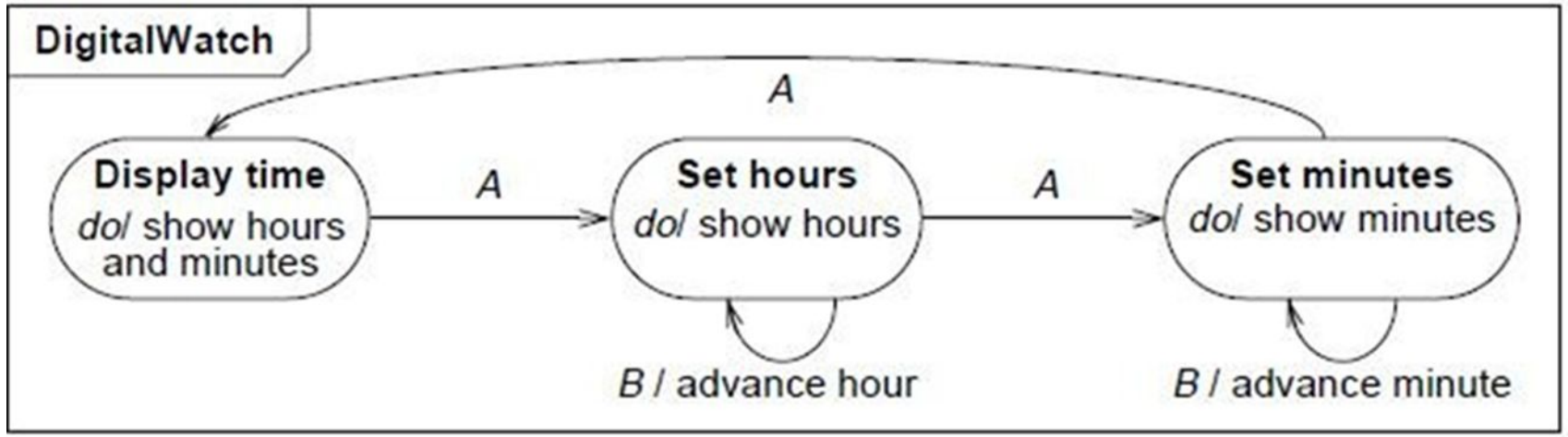


Serving Customer state is a composite state with sequential substates Customer Authentication, Selecting Transaction and Transaction. Customer Authentication and Transaction are composite states by themselves which is shown with hidden decomposition indicator icon. Serving Customer state has triggerless transition back to the Idle state after transaction is finished. The state also has exit action ejectCard which releases customer's card on leaving the state, no matter what caused the transition out of the state.

Source: https://www.uml-diagrams.org/bank-atm-uml-state-machine-diagram-example.html?context=stm-examples

State Diagram: additional Problems

A simple digital watch has a display and two buttons to set it, the A button and the B button. The watch has two modes of operation, display time and set time. In the display time mode, the watch displays hours and minutes, separated by a flashing colon. The set time mode has two sub modes, set hours and set minutes. The A button selects modes. Each time it is pressed, the mode advances in the sequence: display, set hour, set minutes, display, etc. Within the sub modes, the B button advances the hours or minutes once each time it is pressed. Buttons must be released before they can generate another event. Prepare a State diagram of the watch.



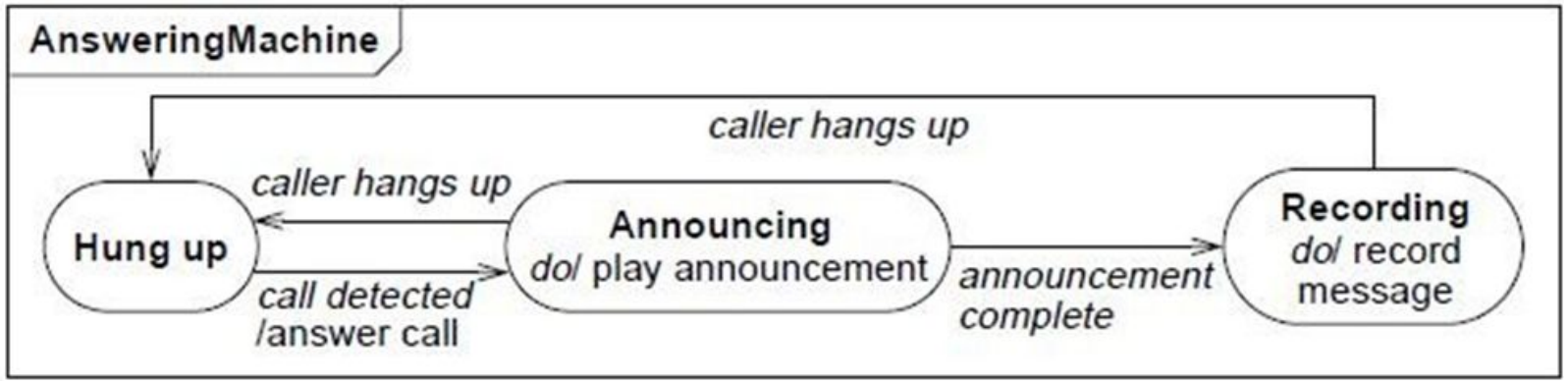
Additional Problem-2:

i) Draw state diagram for the control of a telephone answering machine. The machine detects an incoming call on the first ring and answers the call with a prerecorded announcement. When the announcement is complete, the machine records the caller's message. When the caller hands up, the machine hangs up and shuts off.

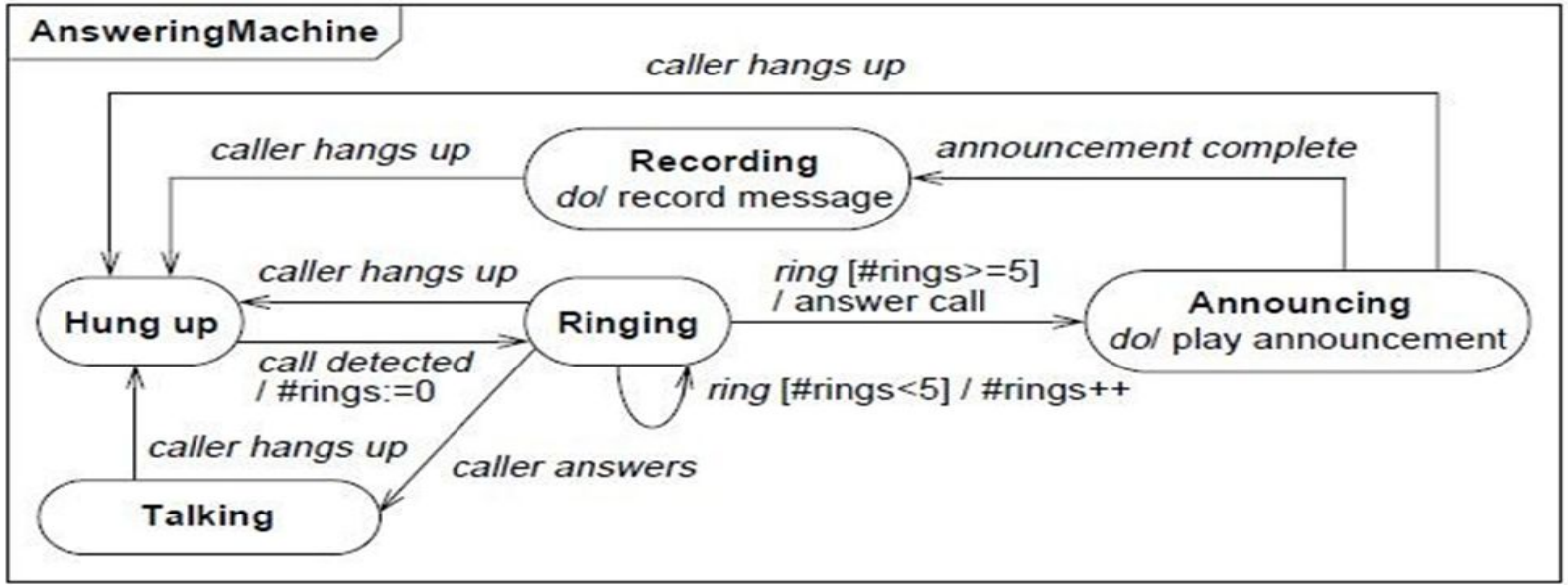
ii) The telephone answering machine in the previous question activates on the first ring. Revise the state diagram so that the machine answers after five rings. if someone answers the telephone before five rings, the machine should do nothing. Be careful to distinguish between five calls in which the telephone is answered on the first ring and one call that rings five times.

Solution–State Diagram

Solution i



Solution ii



References

1. Object - Oriented Modeling and Design With UML by RUMBAUGH and BLAHA , State Modeling
2. UML-diagram.org
3. http://www.ecoa.technology/_static/tutorials/developing/DiningPhilosophers.pdf

THANK YOU

Prof. Sudeepa Roy Dey and Mahitha G

Department of Computer Science and Engineering