

Creating Activity Diagrams

Objectives

Upon completion of this module, you should be able to:

- Identify the essential elements in an Activity diagram
- Model a Use Case flow of events using an Activity diagram

Additional Resources



Additional resources – The following references provide additional information on the topics described in this module:

- Booch, Grady, James Rumbaugh, Ivar Jacobson. *The Unified Modeling Language User Guide*. Reading: Addison Wesley Longman, Inc., 1999.
- Jacobson, Ivar. *Object-Oriented Software Engineering*. Harlow: Addison Wesley Longman, Inc., 1993.
- Rosenberg, Doug, Kendall Scott. *Use Case Driven Object Modeling with UML (A Practical Approach)*. Reading: Addison Wesley Longman, Inc., 1999.
- Rosenberg, Doug, Kendall Scott. *Applying Use Case Driven Object Modeling with UML (An Annotated e-Commerce Example)*. Reading: Addison Wesley Longman, Inc., 2001.
- The Object Management Group. “OMG Unified Modeling Language™ (OMG UML), Superstructure,”
[<http://www.omg.org/spec/UML/2.2/Superstructure/PDF/>],
Version 2.2, February 2009.

Process Map

This module covers the next step in the Requirements Analysis workflow: Creating Activity diagrams. Figure 5-1 highlights the activities and artifacts covered in this module.

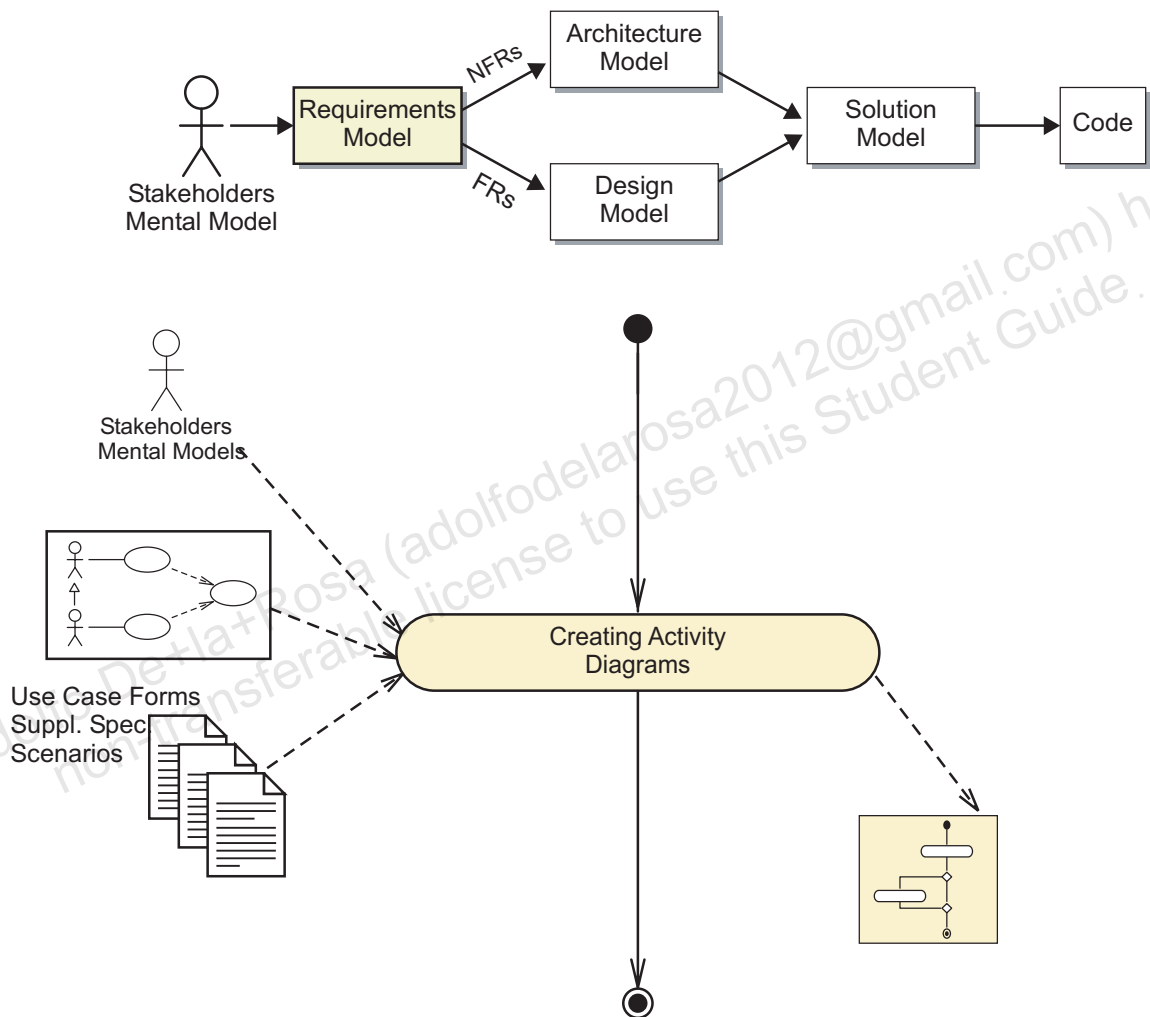


Figure 5-1 Creating Activity Diagrams Process Map

Describing a Use Case With an Activity Diagram

Because use cases are essential to the success of a software project, you should verify that your mental model of the Use Case matches the stakeholder's mental model. You can verify a Use Case by creating a view of your mental model that illustrates the behavior of that Use Case. To do that, you can:

- Model the flow of events of the Use Case in an Activity diagram
- Validate the Use Case by reviewing the Activity diagram with the stakeholders

Activity Diagrams can also be used for the following:

- Model Business Processes
- Model Detailed Design

Business process modeling can include multiple Use Cases and the interaction between actors (human and systems). Whereas a detailed design is often a small part of a Use Case depicted to a fine level of detail. For example, one activity in the analysis diagram may be linked to another diagram showing the internal low level sub activities.

To represent the Use Case in an Activity diagram, you must learn the essential features of these UML diagrams. The essential features of Activity diagrams is presented in the next section.

Identifying the Elements of an Activity Diagram

An Activity diagram represents a sequence of actions (or activities) with a definite beginning and end-point. Figure 5-2 shows the essential elements of an Activity diagram.

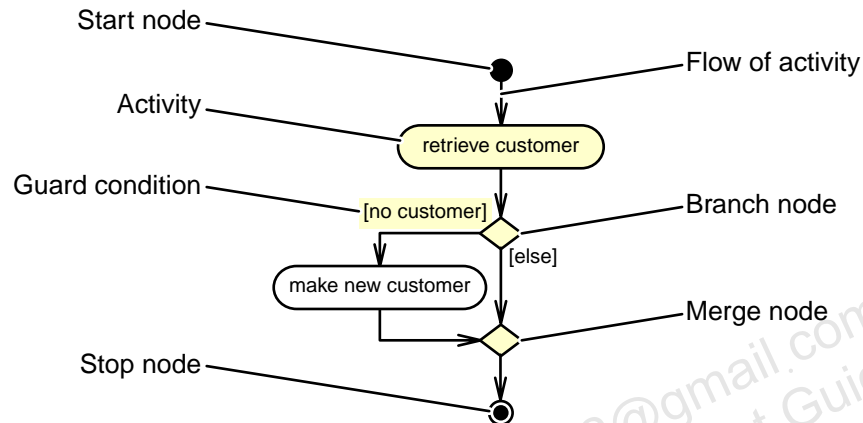


Figure 5-2 Example of an Activity Diagram

Activity diagrams can also represent simultaneous activities in which the activities can be performed in any order. Simultaneous activities are illustrated with fork and join bars. This notation can indicate a truly parallel activity as in the case of a multiprocessor system or it can indicate a set of activities with no specific order. Figure 5-3 illustrates an example for using fork and join bars.

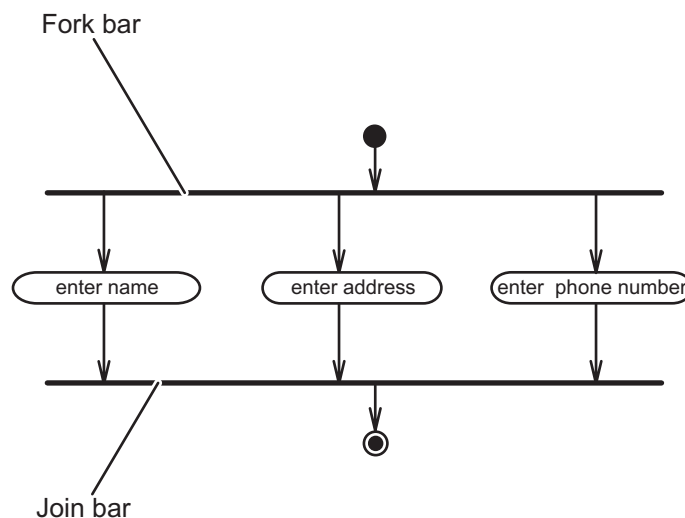


Figure 5-3 An Example of Concurrent Activities

Activities and Actions

Activities and actions are processes taken by the system or an actor.

- Activity nodes and action nodes use the same notation in UML
- An activity can be divided into other activities or actions
- An action is an activity node which cannot be divided within the context of the current view.
- A primitive form of action results in a change in the state of the system or the return of a value.

Flow of Control

An Activity diagram must start with a Start node and end with a Stop node. Flow of control is indicated by the arrows that link the activities together. Figure 5-4 illustrates the flow of control.

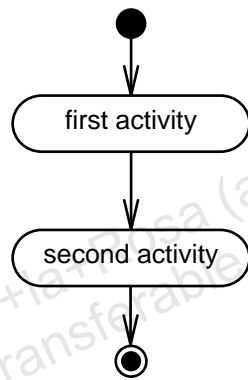


Figure 5-4 An Activity Diagram Represents Flow of Control

Branching

The branch and merge nodes represent conditional flows of activity:

- A branch node has two or more outflows, with Boolean predicates to indicate the selection condition.
- A merge node collapses conditional branches.

Figure 5-5 illustrates branch and merge nodes.

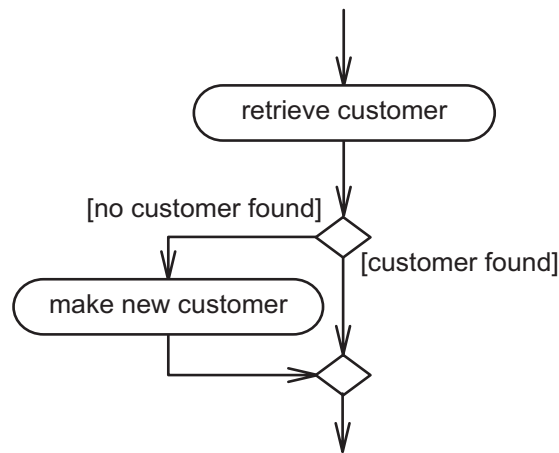


Figure 5-5 Branch and Merge Nodes

Iteration

You can achieve iteration by using branch nodes. Figure 5-6 shows two examples of iteration.

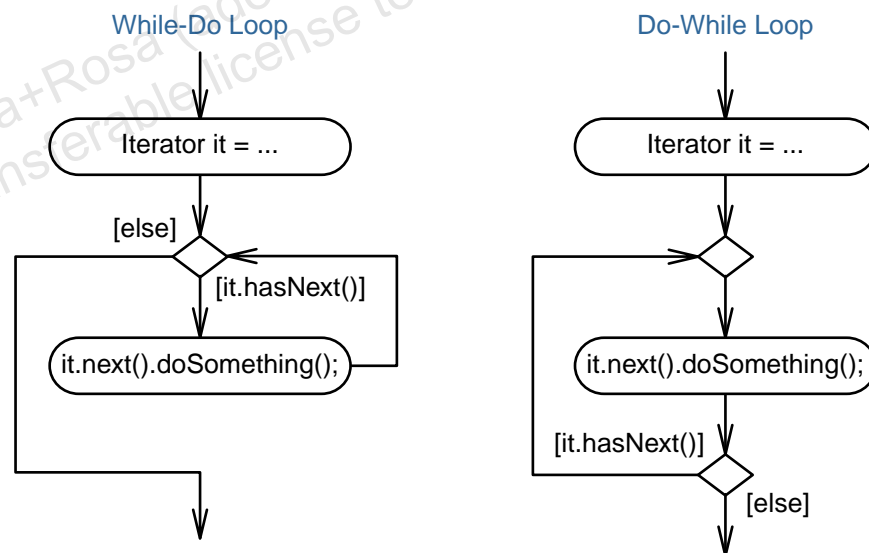


Figure 5-6 Iteration Loops in an Activity Diagram

The first example (on the left) maps closely to a while-do loop in which the loop test is performed before the loop body. The second example (on the right) maps closely to a do-while loop in which the loop body is performed at least once before the loop test is performed.

Concurrent Flow of Control

Use the fork and join bars to indicate concurrent flow of control. Figure 5-7 illustrates this.

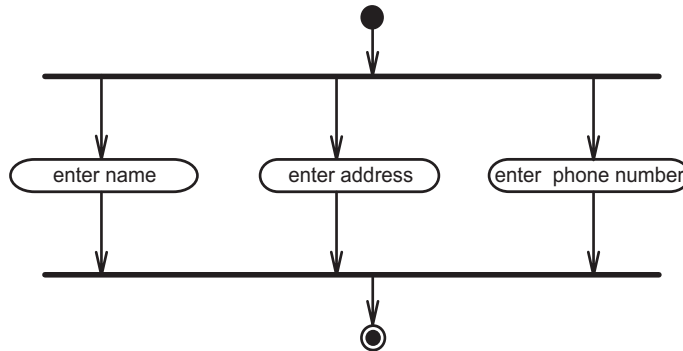


Figure 5-7 Concurrent Flow in an Activity Diagram

- Fork and join bars can represent either threaded activities or parallel user activities.

The fork and join bars can represent two distinct types of behavior:

- fork and join can represent threaded or truly concurrent activities within a computer system.
- fork and join can represent a set of activities that can happen in any, non-specified order.

Passing an Object between Actions

An Activity diagram can show objects being passed between actions

- A pin is a connection point of an action for object input or output.
- The name of the pin denotes the object being passed in or out.

A pin is shown as using a small rectangle attached above or below an activity node.

You can optionally show the state of the object that is being passed within square brackets, next to the object name.

Figure 5-8 illustrates the uses of pins.

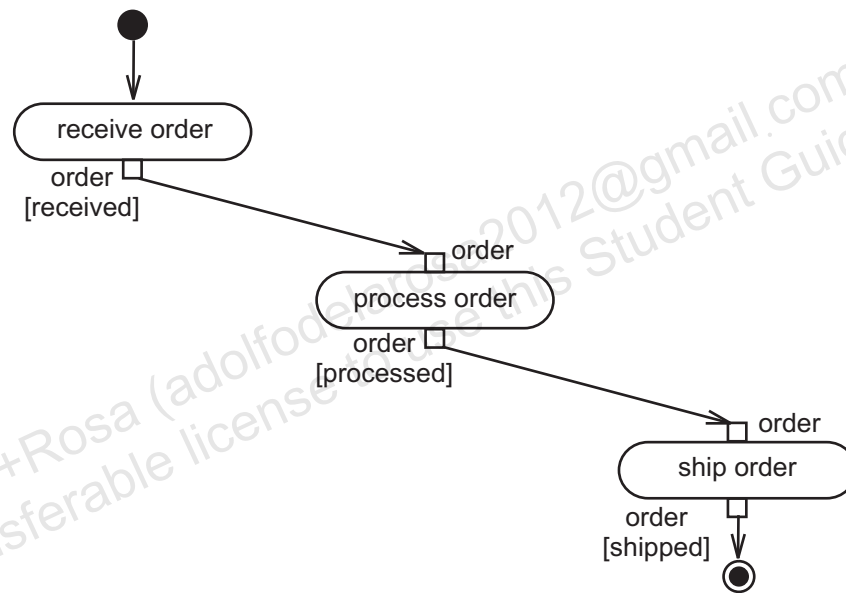


Figure 5-8 Example of Object Flow between Activities

Note – In previous UML versions there were alternative styles used to show the passing of an object. These styles remain valid.



Partitions in Activity Diagrams

An Activity diagram can show objects grouped into partitions (formerly called swimlanes). Partitions are labelled to represent the name of the group. Partitions can be either vertical, horizontal or both vertical and horizontal. Figure 5-9 illustrates an example of vertical partitions.

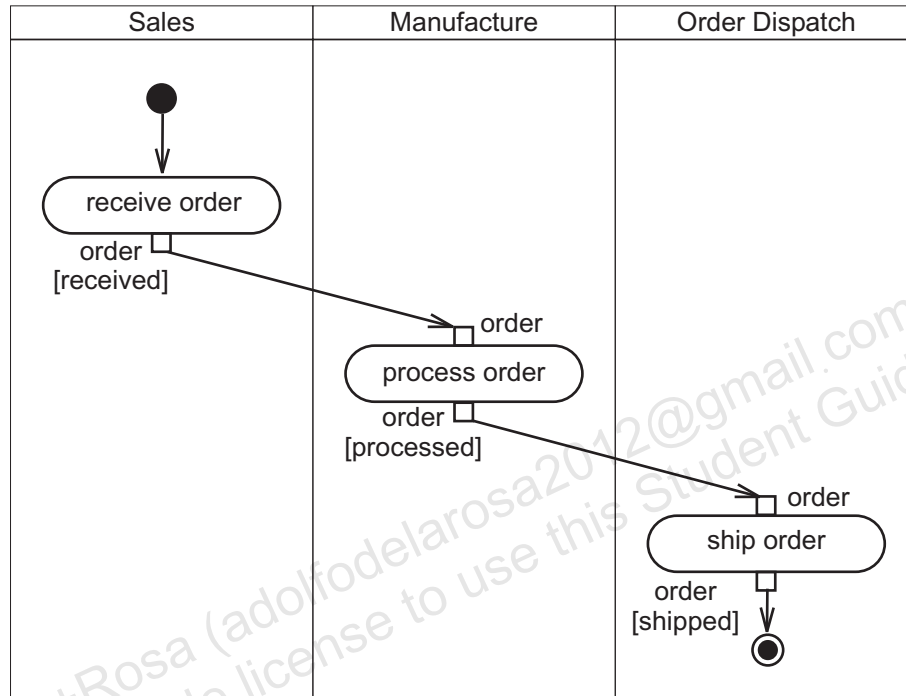


Figure 5-9 Example of Vertical Partitions in an Activity Diagram

Signals in Activity Diagrams

An Activity diagram can show the receiving of a signal, using an:

- Accept Event Action element
- Accept Time Event element

The source of the signal is often external to the current activity diagram. The Time Event is triggered at a specific time or after a time interval.

An Activity diagram can show the sending of a signal, using a Send Event Action element

The destination of the signal is often external to the current activity diagram. Figure 5-10 shows an example where Order Details are presented for confirmation. If the Accept Order signal is received within 10 minutes of the Order Details being presented, then the order is processed. When the Process Order activity is completed the Raise Invoice signal is sent. If the Accept Order signal is not received within 10 minutes of being presented then the Cancel Order activity will commence.

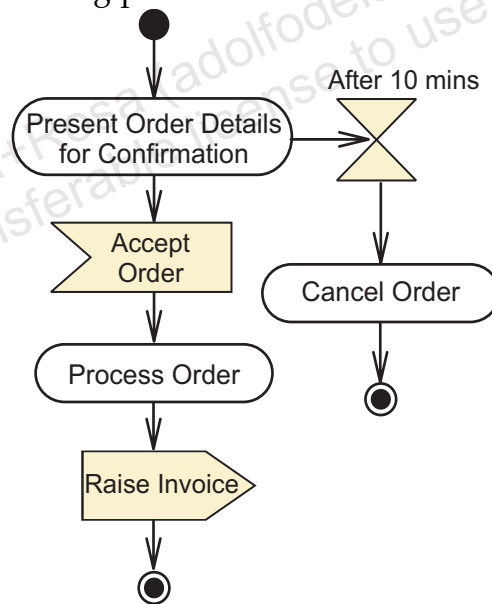


Figure 5-10 Example of Send, Accept and Time Events.

Interruptible Activity Regions

An Activity diagram can show a subset of activities that can be interrupted by an event.

On receipt of an Accept Event or Accept Time Event that occurs within the defined Interruptible Activity region, the current flow stops, and the flow continues from the interrupting event element.

Figure 5-11 shows an example where an order can be cancelled at any time until it is ready to be shipped.

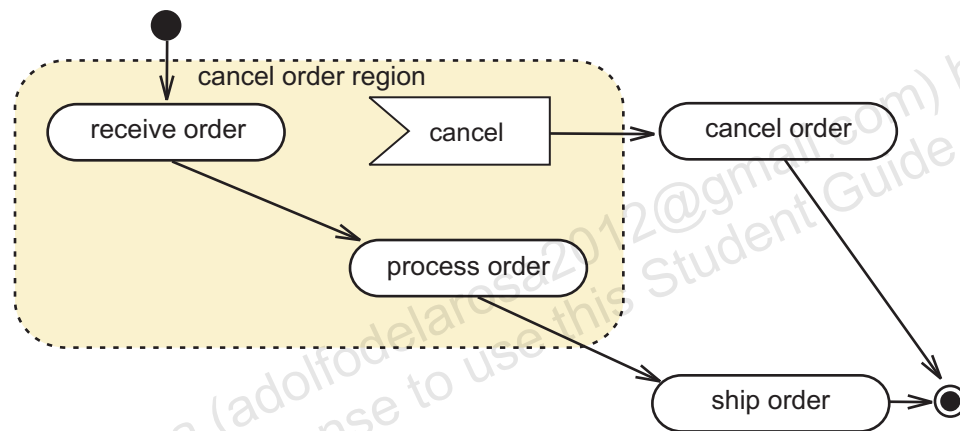


Figure 5-11 Example of an Interruptible Activity Region

Creating an Activity Diagram for a Use Case

To create an Activity diagram for a Use Case, analyze the flow of events field in the Use Case form:

- Identify activities
- Identify branching and looping
- Identify concurrent activities

Figure 5-12 illustrates a simple sequence of activities for part of the Create Reservation Use Case. This diagram shows the activities involved in identifying the customer, by either delegating the entry of the new customer details to the extension point (New Customer), or by the actor entering a subset of customer information in order to find the existing customer. If no existing customer is found then the extension point (New Customer) is used.

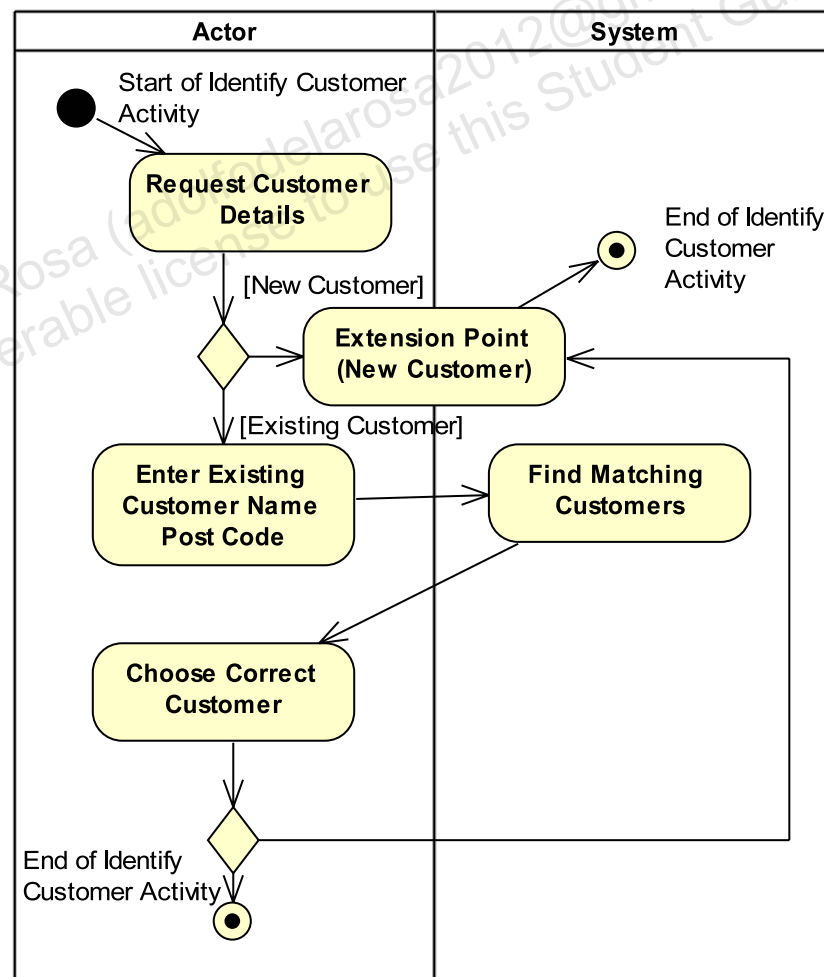


Figure 5-12 Activity Diagram showing a subset of the Create Reservation Use Case

Figure 5-13 shows an Activity Diagram that closely represents the main flow and alternate flow paths of the Create Reservation Use Case Form.

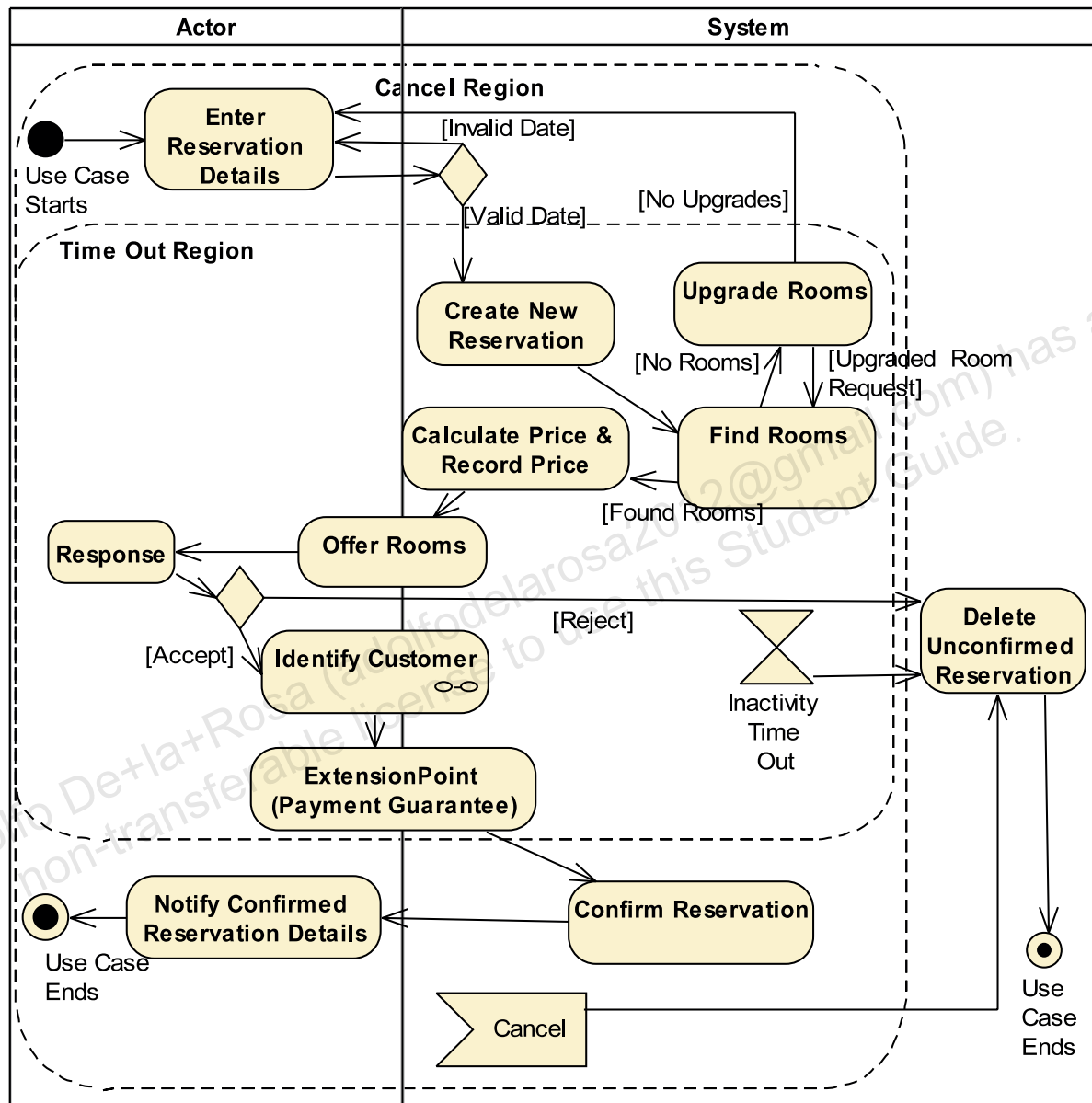


Figure 5-13 Activity Diagram for the Create Reservation Use Case

Summary

This module described:

- The essential elements of an Activity diagram
- How you can visually represent the flow of events of a Use Case with an Activity diagram.
You can present this view of the Use Case to the client-side stakeholders to verify that the development team understands the behavior of the Use Case.

Adolfo De+la+Rosa (adolfodelarosa2012@gmail.com) has a
non-transferable license to use this Student Guide.