UML Dynamic Diagrams

Activity Diagrams

UML : ACTIVITY DIAGRAM

1. Activity diagrams are the object-oriented equivalent of flow charts and data-flow diagrams from structured development.
2. Activity diagrams describe the workflow behavior of a system.
3. The process flows in the system are captured in the activity diagram.
4. Activity diagram illustrates the dynamic nature of a system by modeling the flow of control from activity to activity.

UML : ACTIVITY DIAGRAM

When to Use Activity Diagrams ?

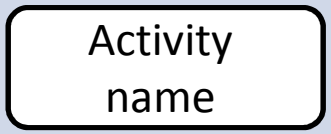
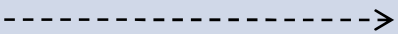
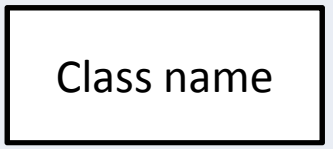
to explore the logic of

1. a complex operation.
2. a complex business rule.
3. a single use case.
4. several use cases.
5. a business process.
6. software processes.

Useful for showing workflow and parallel processing.

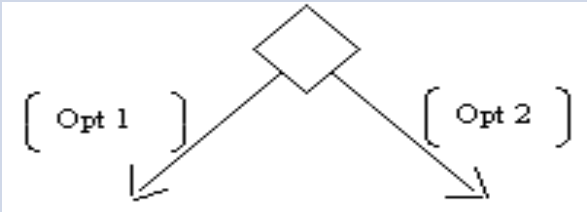
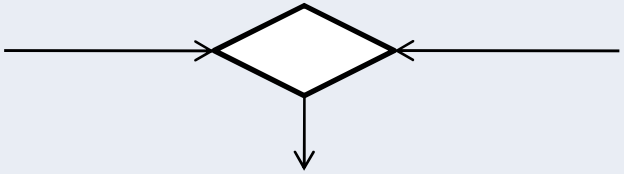
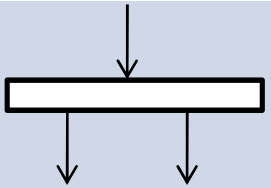
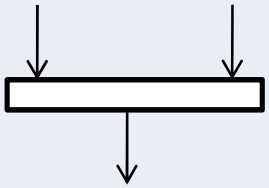
UML : ACTIVITY DIAGRAM

Elements of activity diagram

Description	Symbol
Activity : Is used to represent a set of actions	
A Control Flow: Shows the sequence of execution	
An Object Flow: Shows the flow of an object from one activity (or action) to another activity (or action).	
An Initial Node: Portrays the beginning of a set of actions or activities	
A Final-Activity Node: Is used to stop all control flows and object flows in an activity (or action)	
An Object Node: Is used to represent an object that is connected to a set of Object Flows.	

UML : ACTIVITY DIAGRAM

Elements of activity diagram

Description	symbol
A Decision Node: Is used to represent a test condition to ensure that the control flow or object flow only goes down one path	
A Merge Node: Is used to bring back together different decision paths that were created using a decision-node.	
A Fork Node:Is used to split behavior into a set of parallel or concurrent flows of activities (or actions)	
A Join Node: Is used to bring back together a set of parallel or concurrent flows of activities (or actions).	
A Swimlane :A swimlane is a way to group activities performed by the same actor on an activity diagram or to group activities in a single thread	

UML : ACTIVITY DIAGRAM

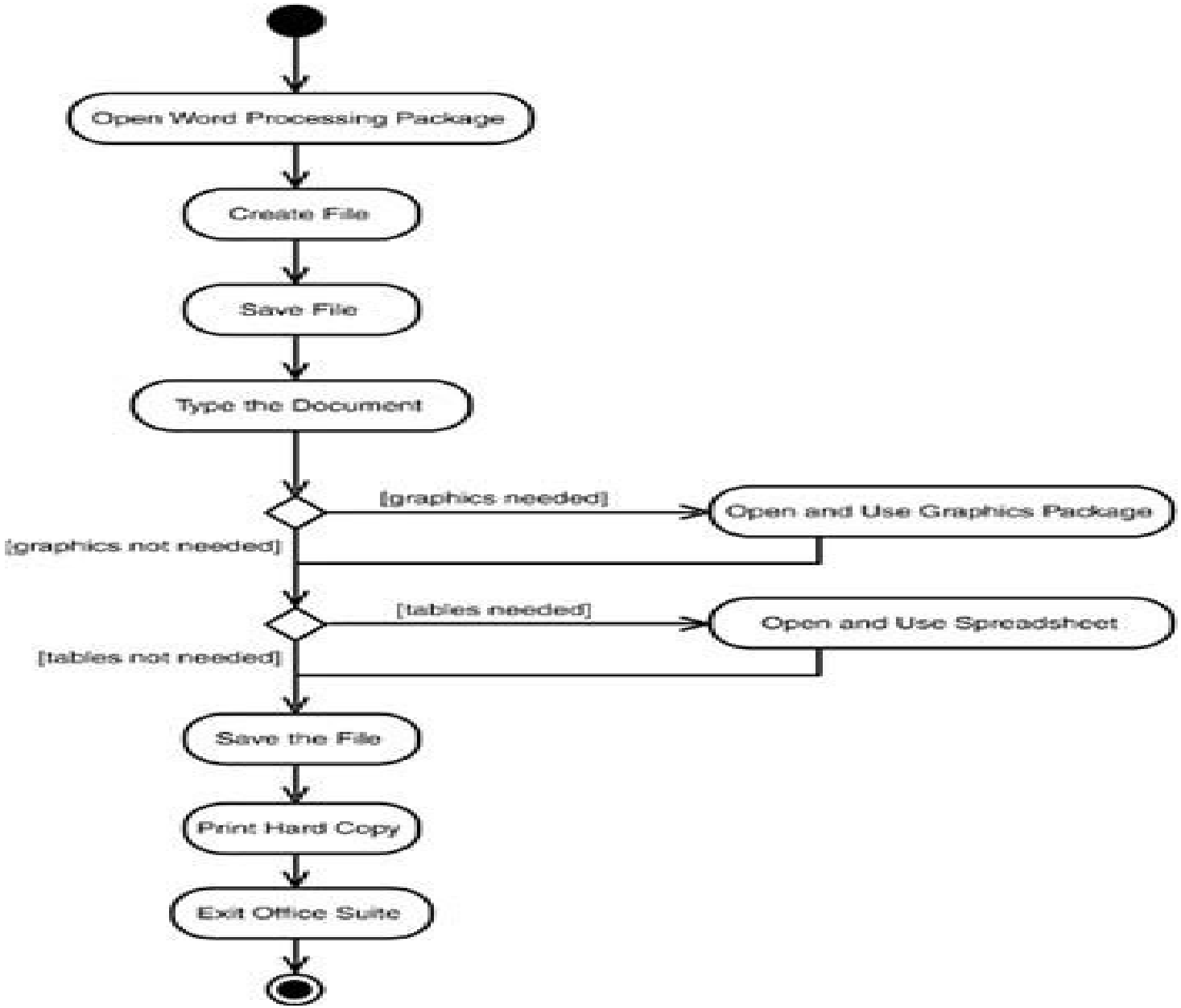
Example 1: creating document .

1. Open the word processing package.
2. Create a file.
3. Save the file under a unique name within its directory.
4. Type the document.
5. If graphics are necessary, open the graphics package, create the graphics, and paste the graphics into the document.
6. If a spreadsheet is necessary, open the spreadsheet package, create the spreadsheet, and paste the spreadsheet into the document.
7. Save the file.
8. Print a hard copy of the document.
9. Exit the word processing package.

UML : ACTIVITY DIAGRAM

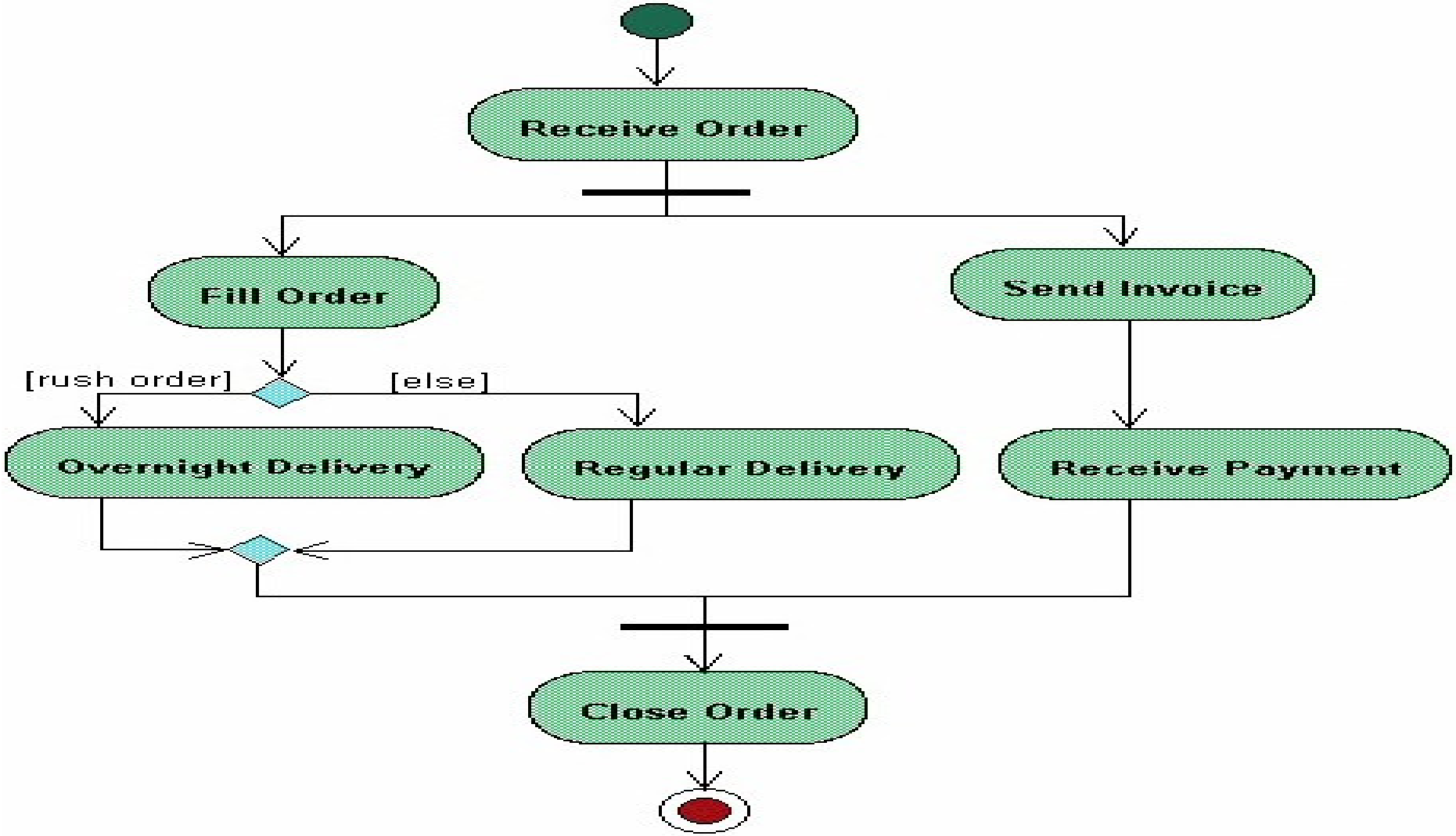
Example 1:

Creating document .



UML : ACTIVITY DIAGRAM

Example 2: processing an order



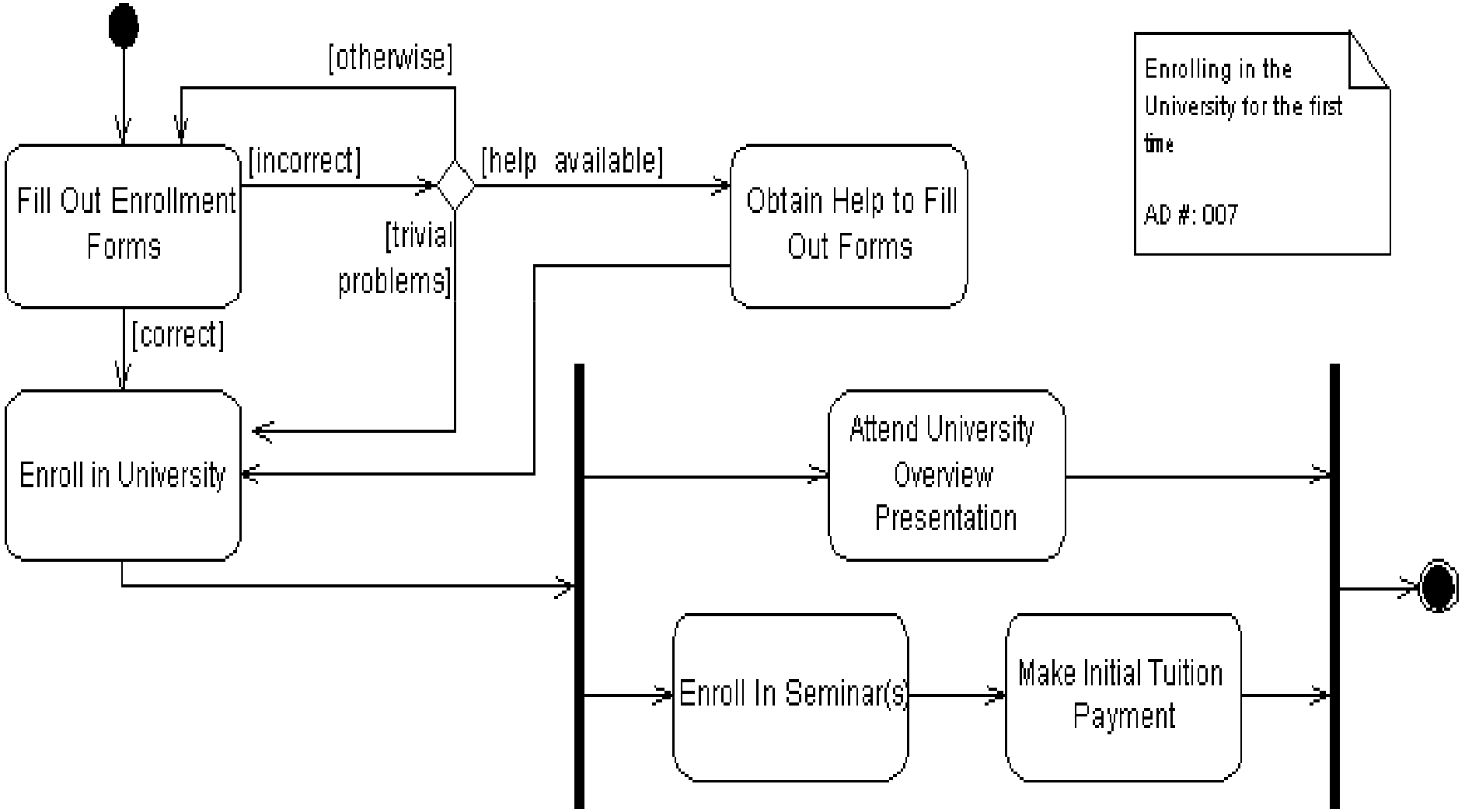
UML : ACTIVITY DIAGRAM

Example 2: processing an order

Once the 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 Fill Order side, the method of delivery is decided conditionally. Depending on the condition either the Overnight Delivery activity or the Regular Delivery activity is performed. Finally the parallel activities combine to close the order.

UML : ACTIVITY DIAGRAM

Example 3: enrollment in university



UML : ACTIVITY DIAGRAM

Example 3: Example 3: enrollment in university

1. An applicant wants to enroll in the university.
2. The applicant hands a filled out copy of form *U113 University Application Form* to the registrar.
3. The registrar inspects the forms.
4. The registrar determines that the forms have been filled out properly.
5. The registrar informs student to attend in university overview presentation.
6. The registrar helps the student to enroll in seminars
7. The registrar asks the student to pay the initial.

UML : ACTIVITY DIAGRAM

Activities guideline:

1. Question “Black Hole” Activities. A black hole activity is one that has transitions into it but none out, typically indicating that you have either missed one or more transitions.
2. Question “Miracle” Activities. A miracle activity is one that has transitions out of it but none into it, something that should be true only of start points.

UML : ACTIVITY DIAGRAM

Guards

A guard is a condition that must be true in order to traverse a transition.

1. Each Transition Leaving a Decision Point Must Have a Guard . This ensures that you have thought through all possibilities for that decision point.
2. Guards Should Not Overlap. For example guards such as x <0, x = 0, and x > 0 are consistent whereas guard such as x <= 0 and x >= 0 are not consistent because they overlap – it isn't clear what should happen when x is 0.
3. Guards on Decision Points Must Form a Complete Set. For example, guards such as x < 0 and x >0 are not complete because it isn't clear what happens when x is 0.

UML : ACTIVITY DIAGRAM

Guards

A guard is a condition that must be true in order to traverse a transition.

4. Exit Transition Guards and Activity Invariants Must Form a Complete Set. An activity invariant is a condition that is always true when your system is processing an activity.
5. Apply a [Otherwise] Guard for "Fall Through" Logic. In example3 , you can see that one of the transitions on the decision point is labeled *Otherwise*, a catchall condition for the situation in which problems with the forms are not trivial and help is not available. This avoided a very wordy guard, thus simplifying the diagram.
6. Guards Are Optional. It is very common for a transition to not include a guard, even when an activity includes several exit transitions. Follow Agile Modeling (AM)'s principle of Depict Models Simply and only indicate a guard on a transition if it adds value.

UML : ACTIVITY DIAGRAM

Parallel Activities guidelines

1. It is possible to show that activities can occur in parallel, as you see in example3 depicted using two parallel bars. The first bar is called a fork, it has one transition entering it and two or more transitions leaving it. The second bar is a join, with two or more transitions entering it and only one leaving it.
2. A Fork Should Have a Corresponding Join. In general, for every start (fork) there is an end (join). In UML 2 it is not required to have a join, but it usually makes sense.
3. Forks Have One Entry Transition.
4. Joins Have One Exit Transition
5. Avoid Superfluous Forks.

UML : ACTIVITY DIAGRAM

Swimlane Guidelines

A swimlane is a way

To group activities

Performed by the

Same actor on an

Activity diagram or

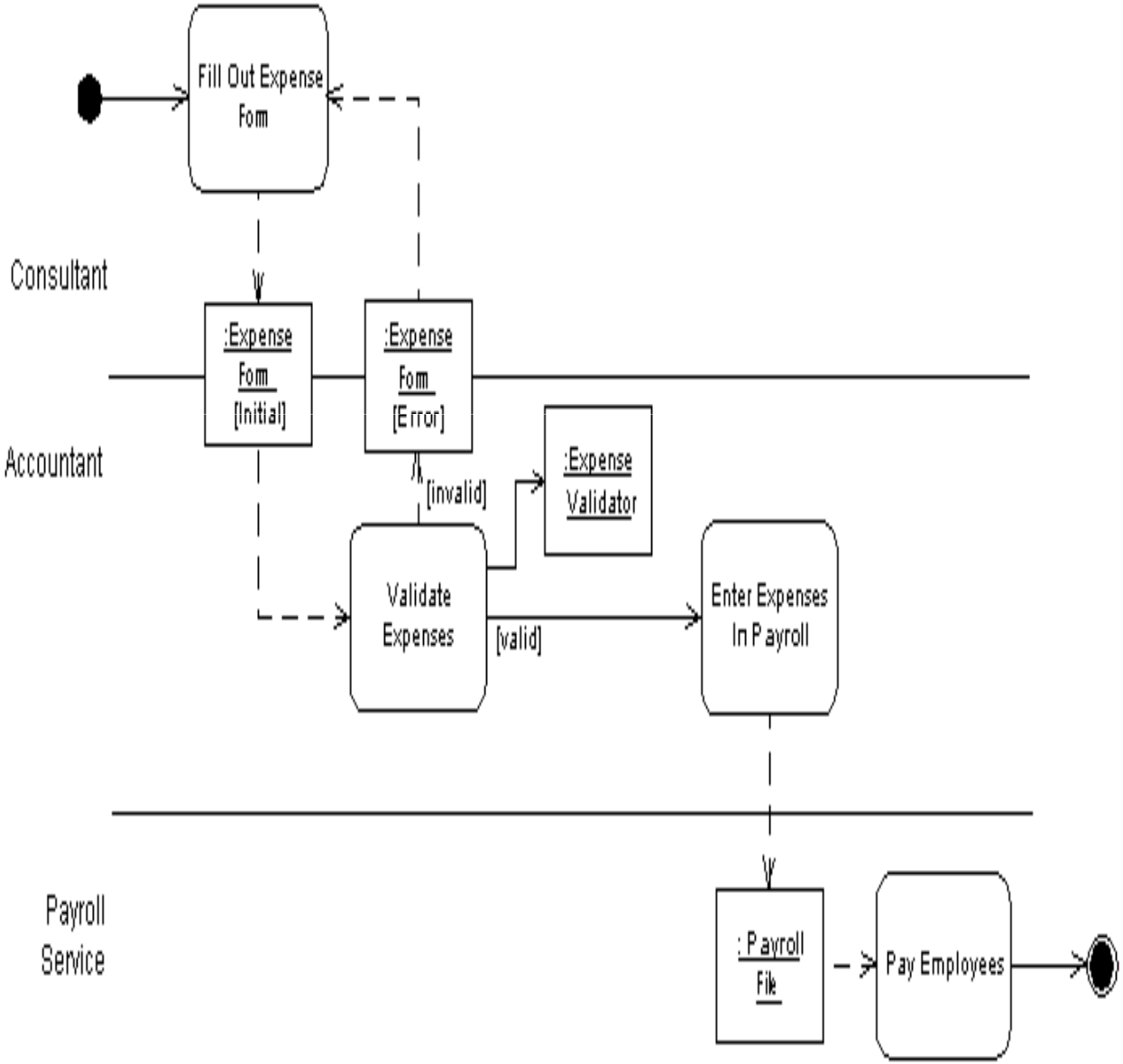
to group activities

in a single thread

Example 4:

Submitting

expenses



UML : ACTIVITY DIAGRAM

Swimlane Guidelines

1. Order Swimlanes in a Logical Manner.
2. Apply SwimLanes To Linear (sequential) Processes. A good rule of thumb is that swimlanes are best applied to linear processes.
3. Have Less Than Five Swimlanes.
4. Consider Swimareas For Complex Diagrams.
5. SwimLane Suggest The Need to Reorganize Into Smaller Activity Diagrams.
6. Consider Horizontal Swimlanes for Business Processes. Left to right “west culture”.

UML : ACTIVITY DIAGRAM

Example 5: business process of meeting a new client

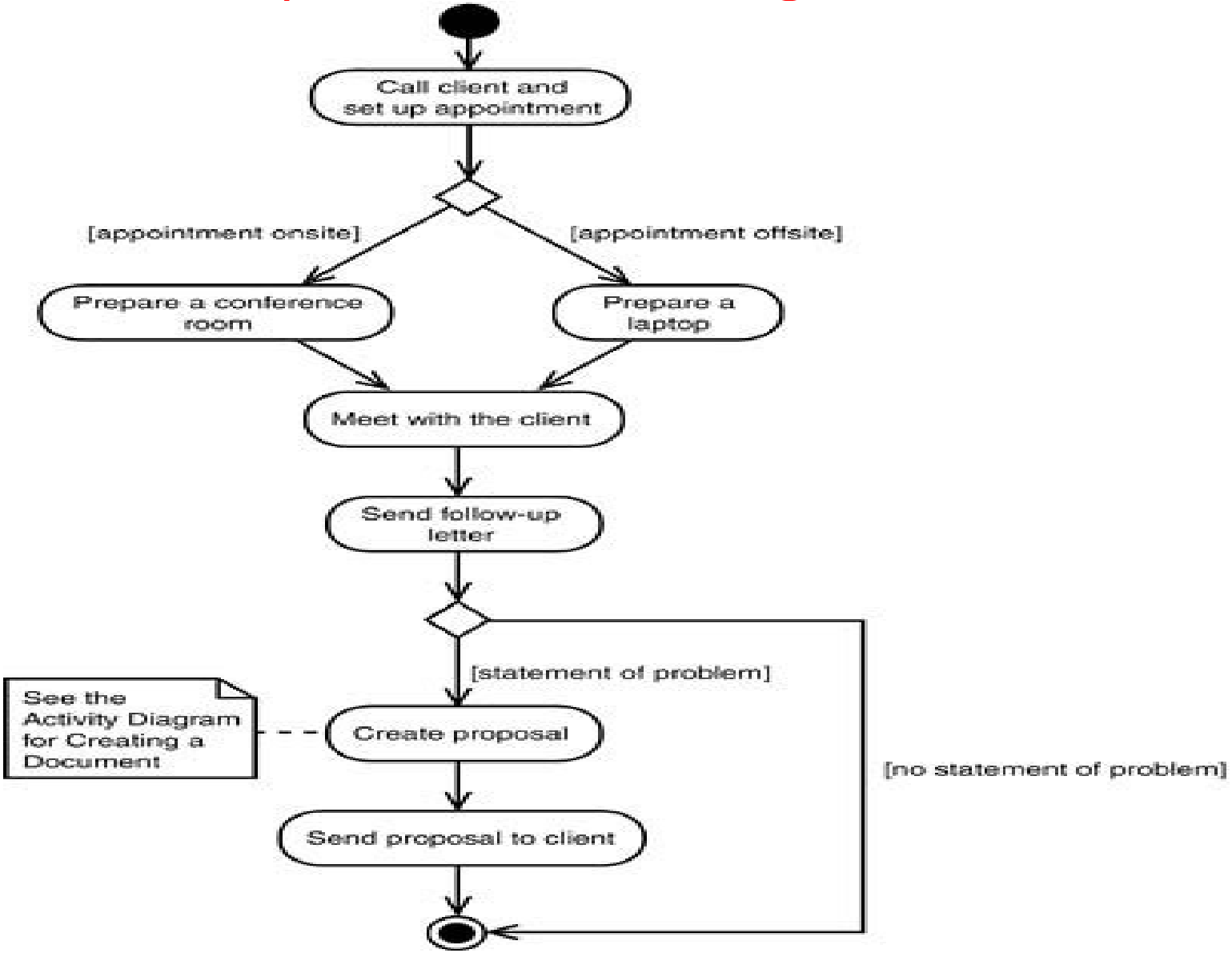
1. A salesperson calls the client and sets up an appointment.
2. If the appointment is onsite (in the consulting firm's office), corporate technicians prepare conference room for a presentation
3. If the appointment is offsite (at the client's office), a consultant prepares a presentation on a laptop.
4. The consultant and the salesperson meet with the client at the agreed-upon location and time.
5. The salesperson follows up with a letter
6. If the meeting has resulted in a statement of a problem, the consultant create a proposal and sends it to the client.

UML : ACTIVITY DIAGRAM

Example 5: business process of meeting a new client

Without Swimlane for Create a Proposal

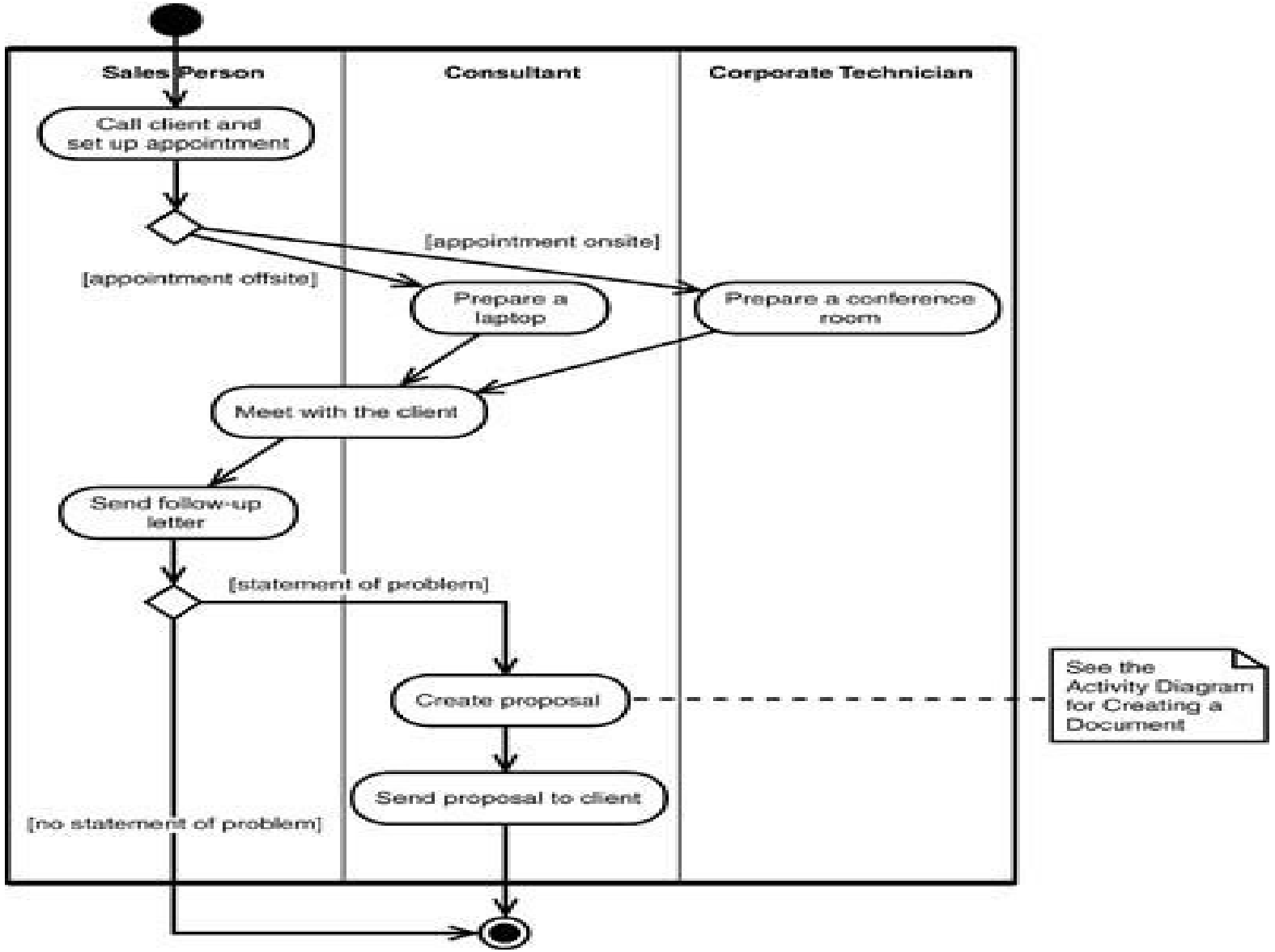
See example1



UML : ACTIVITY DIAGRAM

Example 5: business process of meeting a new client

With a swimlane



UML : ACTIVITY DIAGRAM

Action-Object Guidelines

1. Activities act on objects, For example in example4 the ExpenseForm action object is likely a paper form.
2. Place Shared Action Objects on Swimlane Separators
3. When An Object Appears Several Time Apply State Names. The ExpenseForm object appears twice on the diagram-an initial version of it and one with errors. To distinguish between them, their state names-in this case Initial and Error -are indicated using the same notation as for guards on transitions
4. State Names Should Reflect the Lifecycle Stage of an Action Object
5. Show Only Critical Inputs and Outputs
6. Depict Action Objects As Smaller Than Activities

UML : ACTIVITY DIAGRAM

Action-Object Guidelines

1. State Names Should Reflect the Lifecycle Stage of an Action Object .
2. Show Only Critical Inputs and Outputs
3. Depict Action Objects As Smaller Than Activities

The focus of a UML activity diagram is activities, not the actions implementing or being produced by those activities. Therefore, you can show this focus by having larger activity symbols. To depict the fact that an activity is implemented by an action object, you use a solid arrow. In example4 the ExpenseValidator object implements the Validate Expenses activity.

UML : ACTIVITY DIAGRAM

Homework

Draw an activity diagram for the following problem:

Appointment system for doctor office.

1. A patient came to office, the scheduler get patient info.
2. If the patient is new the scheduler make new patient record.
3. The scheduler display list of possible appointments to patient.
4. Patient choose new appointments , modify appointments or cancel his appointments .
5. Patient make payment.

UML : ACTIVITY DIAGRAM

Homework

Draw an activity diagram for the following problem:

Appointment system for doctor office.

Hints:

There are about 6 to 8 activities and 2 to 5 objects.

UML : ACTIVITY DIAGRAM

References:

1. Systems Analysis & Design with UML, 2nd Edition.
2. Elements of UML 2.0 style.
3. Software Engineering - Sams - Teach Yourself Uml In 24 Hours.