

5-Business Process Modelling with Activity Diagram

IT 3106– Object Oriented Analysis and Design

Level II - Semester 3

Overview

In this section students will

- Be introduced to Activity Diagrams with their benefits
- Learn Components of a Activity Diagram, and steps involved in drawing the diagram
- Learn to Verify and Validate the Business Processes and Functional Models

Intended Learning Outcomes

At the end of this lesson students will be able to

- Able to describe the benefits of using Business Process Modelling with Activity Diagram
- Able to model Use-Case activities using Activity Diagrams
- Able to Identify and describe the steps for preparing an Activity diagram
- Able to Identify and draw swim lanes, synchronization bars in activity diagrams
- Able to draw an Activity Diagram for a given Scenario

List of sub topics

5. Business Process modeling with Activity Diagrams (3 hours)

5.1 Introduction to Activity Diagrams [Ref 1: Pg. 129-130]

5.2 Elements of an Activity Diagram [Ref 1: Pg. 131-136]

5.2.1 Actions and Activities

5.2.2 Object Nodes

5.2.3 Control Flows and Object Flows

5.2.4 Control Nodes (Initial, final-activity, final-flow, decision, merge, fork, and join)

5.1.4 Swimlanes

5.3 Creating Activity Diagrams [Ref 1: Pg. 136-140]

5.4 Verifying and Validating the Business Processes and Functional Models [Ref 1: Pg. 153-157]

Ref 1: Alan Dennis, Barbara Haley, David Tegarden, Systems analysis design, An Object Oriented Approach with UML : an object oriented approach, 5th edition, John Wiley & Sons, 2015, ISBN 978-1-118-80467-4

5.1 Introduction to Activity Diagrams

- Models the process steps or Use Case activities of the system.
- They represents the dynamics of a system.
- They are similar to flow charts.
- They graphically show the work flow of a system.

5.1 Introduction to Activity Diagrams

- They show the flow of control from activity to activity in the system.
- They show what activities can be done in parallel, (That is where they are different from flowcharts.)
- They also show alternative paths through the flow.
- used to model the behavior in a business process independent of objects. Hence they can be used with Non object oriented approaches.

5.1 Introduction to Activity Diagrams

- After creating a use case diagram, *activity diagrams* may be created to represent the flow ***across use cases*** or they may be created to represent the ***flow within a particular use case***.
- Activity diagrams may also be created to show the ***workflow for an operation***.

5.2 Elements of an Activity Diagrams

Activity diagrams contain :

- **Activities,**
- **Transitions** between activities,
- **Decision points,**
- **Synchronization bars.**
- **Swimlanes**

5.2 Elements of an Activity Diagrams

Activity



- Represented in UML by a rounded rectangle.
- Activity represents the performance of some behavior in the work flow.

Transitions



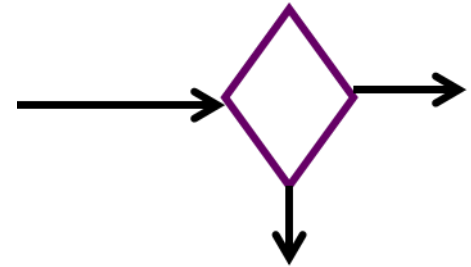
- Transitions are used to show the passing of the flow of control from activity to activity.
- They are typically triggered by the completion of the behavior in the originating activity.

Decision Points



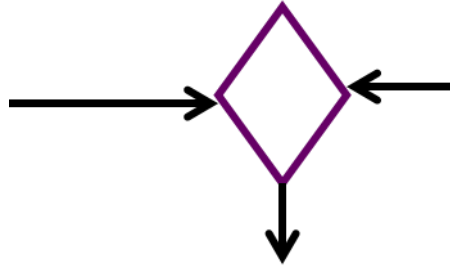
- When modeling the workflow of a system, it is often necessary to show where the flow of control branches based on a decision point.
- The transition from a decision point contain a guard condition.

Decision Points cont..



- The guard condition is used to determine which path from the decision point is taken.
- Decisions along with their guard conditions allow you to show alternative paths through a work flow.

Merge Point

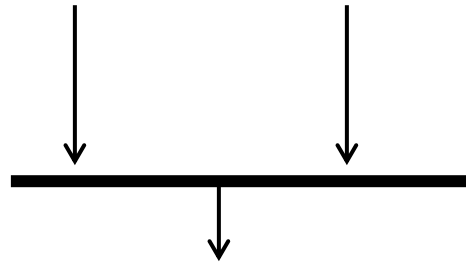


- Two or more flows come in and one flow goes out.
- This combines flows that were previously separated by decisions.
- Processing continues with any one flow coming into the merge.

Synchronization Bars

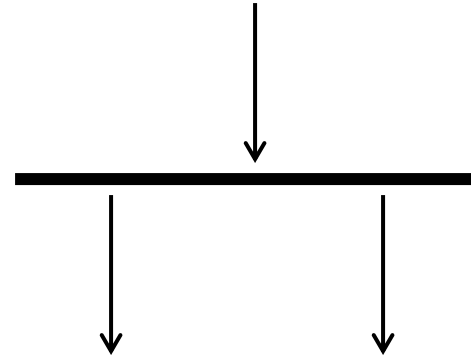
- In a workflow there are typically some activities that may be done in parallel.
- A synchronization bar allows you to specify what activities may be done concurrently.

Join



- Synchronization bars used to show joins in the workflow.
- ie. What activities must complete before processing may continue.
- All actions coming into the join must be completed before processing continues.

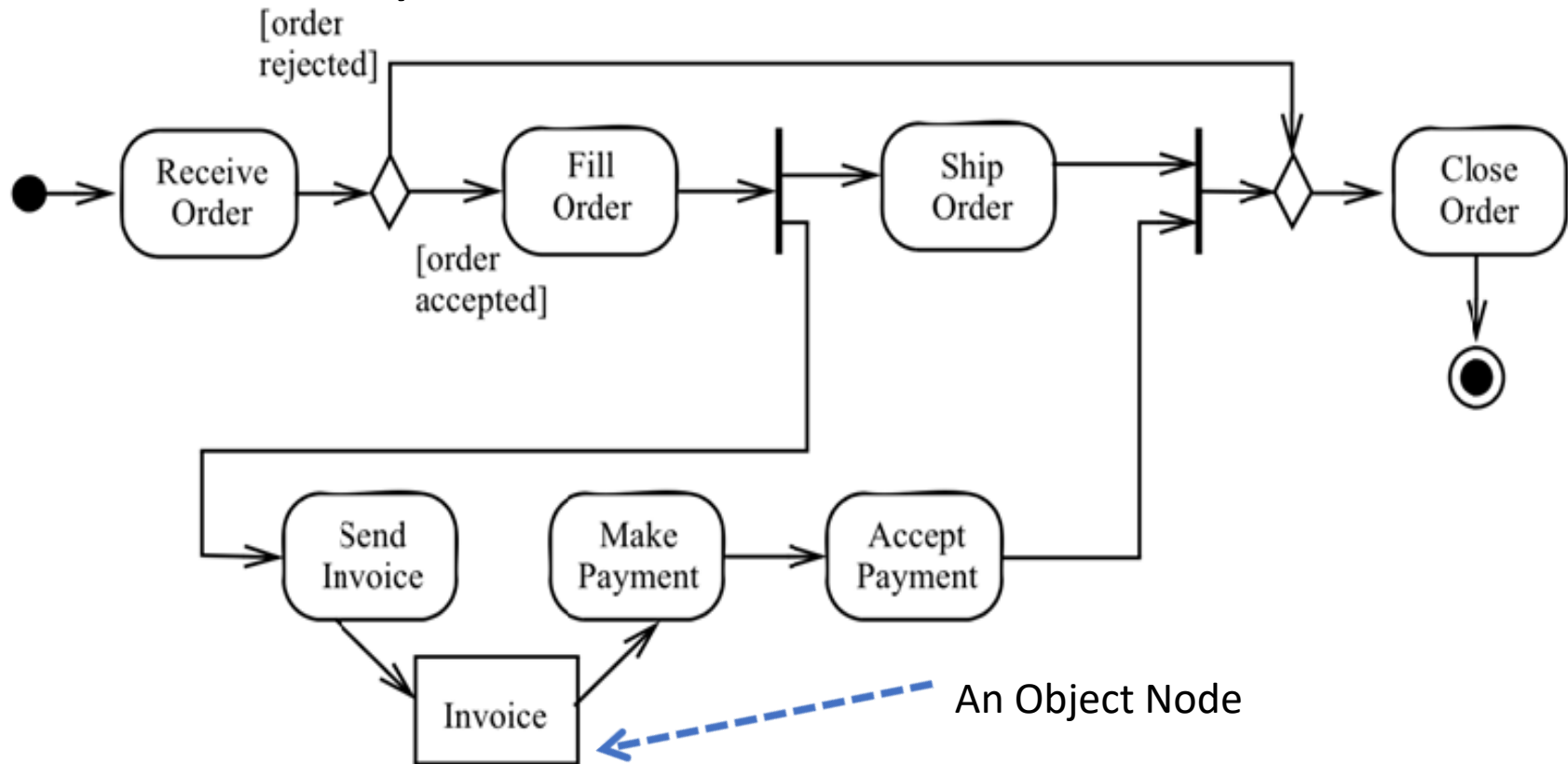
Fork



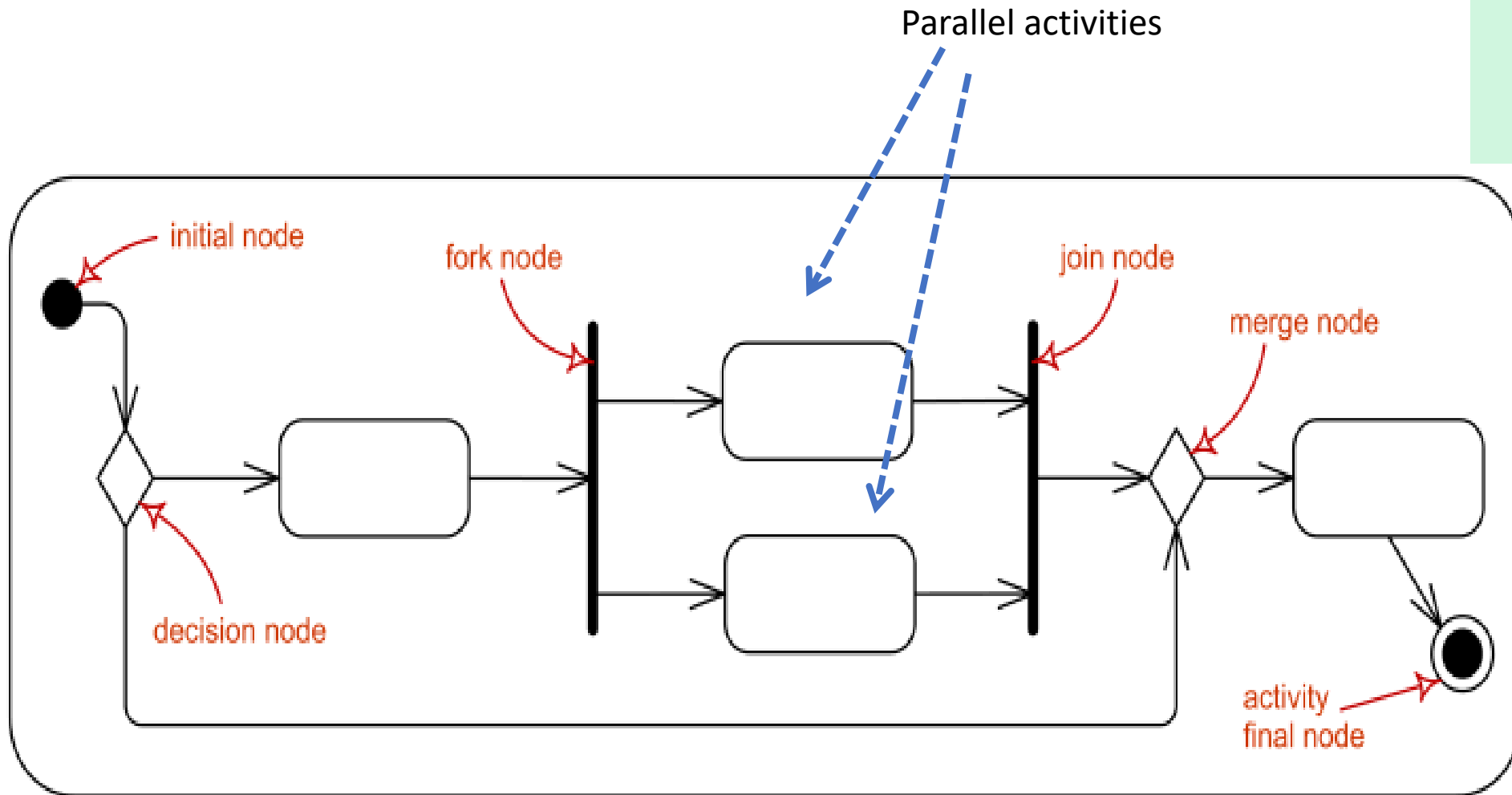
- Synchronization bars can also be used to show forks in the workflow.
- Actions on parallel flows beneath the fork can occur in any order or concurrently.

An object node

- Is used to represent an object that is connected to a set of object flows.
- Is labeled by its class name.



Activity control nodes overview.



Swim lanes

- *Swim lanes* may be used to partition an activity diagram.
- This facility allows activity diagrams to expand and show who has the responsibility for each activity in a process.

Swim lanes cont...

Eg:- Consider a consulting firm and the business process involved in meeting a new client;

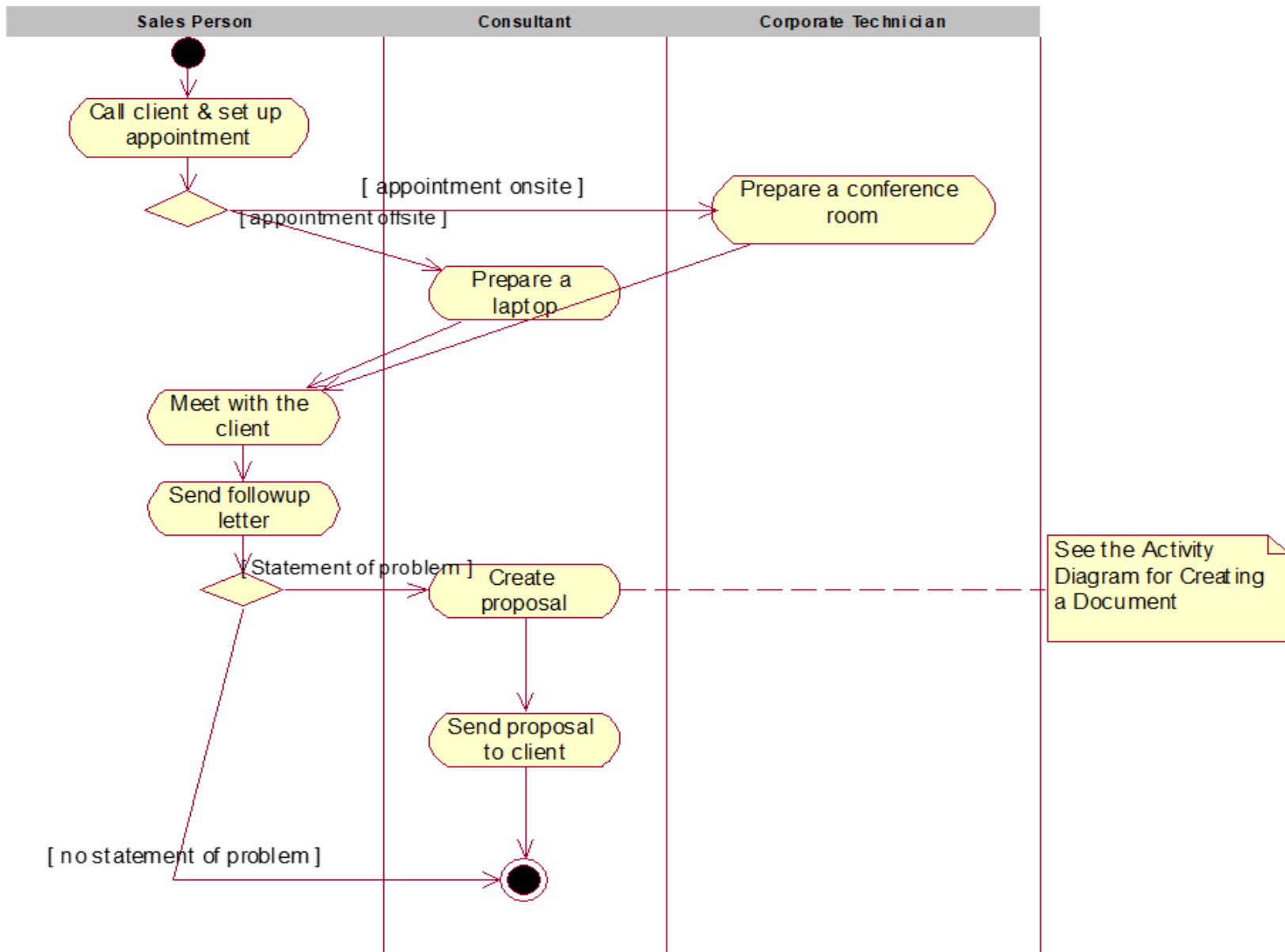
Following are the activities:

- A salesperson calls the client and sets up an appointment.
- If the appointment is onsite, corporate technicians prepare a conference room for a presentation.

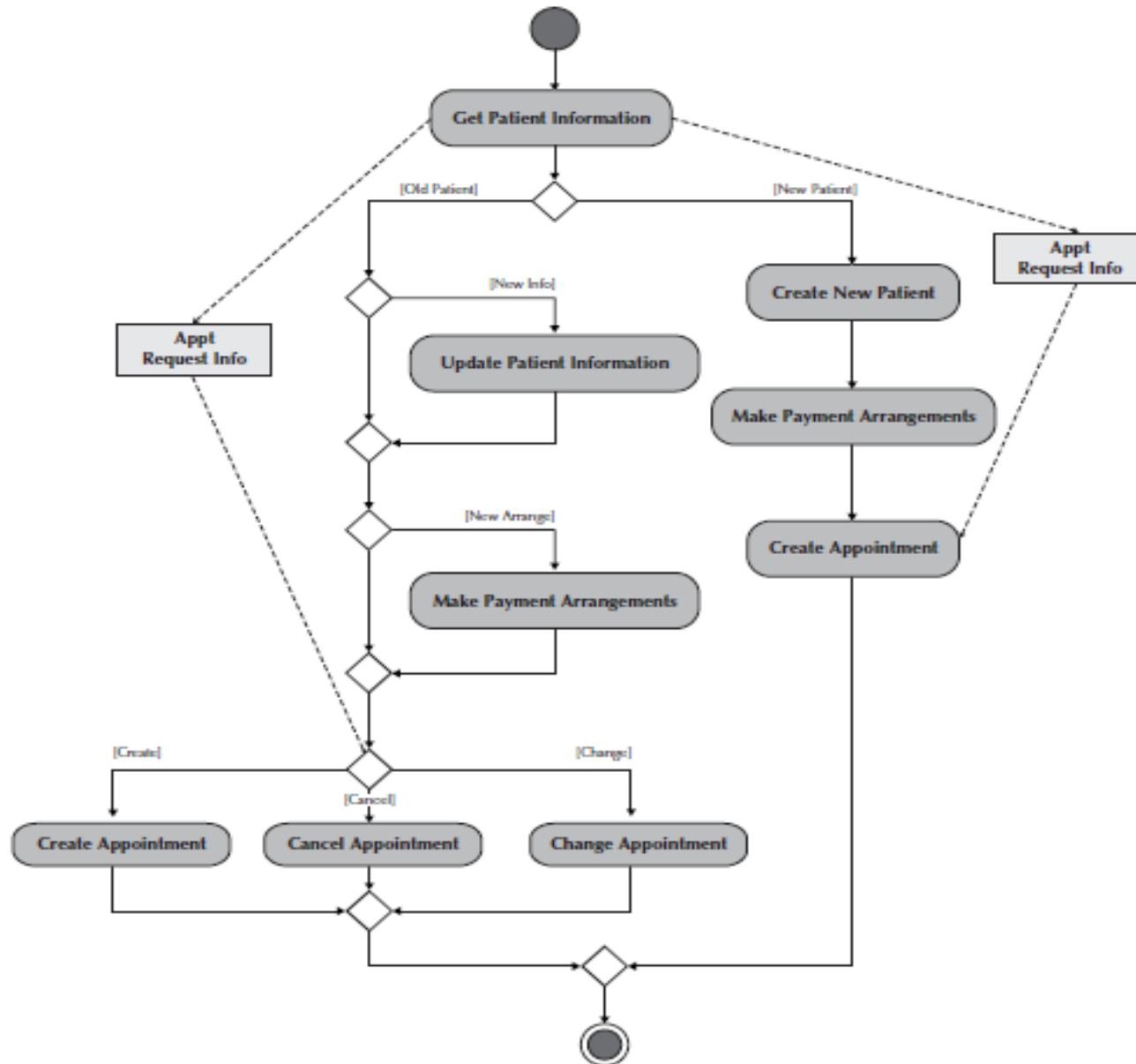
Swim lanes cont...

- If the appointment is offsite, a consultant prepares a presentation on a laptop.
- The consultant and the salesperson meet with the client at the agreed-upon location and time.
- The salesperson follows up with a letter.
- If the meeting has resulted in a statement of a problem, the consultant creates a proposal and sends it to the client

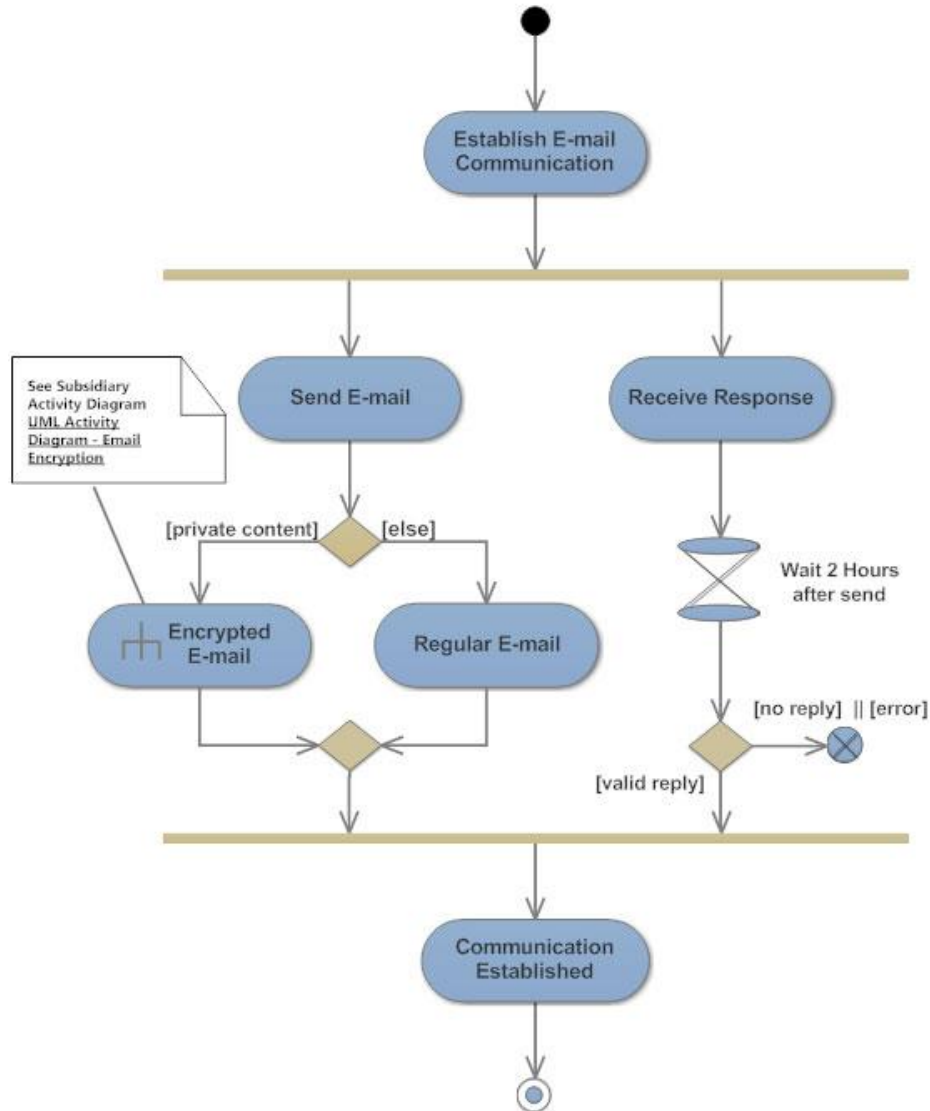
An Activity Diagram



Eg. Activity diagram for Manage appointment Use Case



UML Activity Diagram: Email Connection



New concepts included in UML 2.0

Passage of time =>

Eg. Accept time event



3-6 Min

Eg. Action Node



Wait 10 seconds

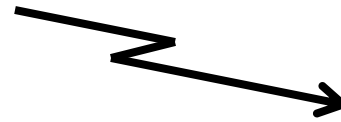


Send company tax return

End of business year occurred

New concepts included in UML 2.0 cont...

- Exception activity
 - Use to show an interrupting activity.



- Subactivity indicator
 - Action invokes another activity



5.3 Creating Activity Diagrams

- There are five steps in creating an activity diagram to document and model a business process.
 1. choose a business process that was previously identified to model.
 2. identify the set of activities necessary to support the business process.
 3. identify the control flows and nodes necessary to document the logic of the business process.
 4. identify the object flows and nodes necessary to support the logic of the business process.
 5. layout and draw the activity diagram to document the business process.

5.4 Verifying and Validating the Business Processes and Functional Models

- We need to verify and validate the current set of functional models to ensure that they faithfully represent the business processes under consideration.
- for example, we must be sure that the activity diagram(s), use-case descriptions, and use-case diagrams all describe the same functional requirements.
- Before we describe the specific tests to consider, we describe walkthroughs, a manual approach that supports verifying and validating the evolving models.

Verification and Validation through Walkthroughs

- A walkthrough which is a review of the different models and diagrams created during functional modeling.
- This interactive review typically is completed by a team whose members come from the development team and the client.
- The purpose of a walkthrough is to thoroughly test the consistency of the functional models to the functional requirements.
- a walkthrough uncovers errors or faults in the evolving specification. However, a walkthrough does not correct errors—it simply identifies them. Error correction is to be accomplished by the team after the walkthrough is completed.

Verification and Validation through Walkthroughs cont....

Typical members of a walkthrough are :

- the presenter : should be played by the person who is primarily responsible for the specific representation being reviewed. This individual presents the representation to the walkthrough team.
- recorder, or scribe: should be a member of the analysis team. This individual carefully takes the minutes of the meeting by recording all significant events that occur during the walkthrough. In particular, all errors that are uncovered must be documented so that the analysis team can address them.
- Another role is to have someone who raises issues regarding maintenance of representation. Owing to the emphasis on reusability in object-oriented development, this role becomes particularly crucial.
- And someone must be responsible for calling, setting up, and running the walkthrough meetings.

Functional Model Verification and Validation

- suggested three different representations for the functional model:

- ❖ activity diagrams,
- ❖ use-case descriptions,
- ❖ and use-case diagrams.

- These three representations should be consistent among themselves.

Eg- When comparing an activity diagram to a use-case description there should be at least one event recorded in the normal flow of events, sub flows, or alternative/exceptional flows of the use-case description for each activity or action that is included on an activity diagram, and each event should be associated with an activity or action.

- Objects portrayed as an object node in an activity diagram must be mentioned in an event in the normal flow of events, sub flows, or alternative/exceptional flows of the use-case description.
- Sequential order of the events in a use-case description should occur in the same sequential order of the activities contained in an activity diagram.

Summary

- Models the process steps or Use Case activities of the system and graphically show the work flow of a system.
- After creating a use case diagram, activity diagrams may be created to represent
 - the flow across use cases or
 - the flow within a particular use case.
 - to show the workflow for an operation.
- Activity diagrams contain activities, transitions between activities, decision points, synchronization bars and swimlanes.
- Synchronization bars used to show *joins* in the workflow which shows what activities must complete before processing may continue.
- Synchronization bars can also be used to show *forks* in the workflow. Actions on parallel flows beneath the *fork* can occur in any order or concurrently

Summary cont..

- Swim lanes may be used to partition an activity diagram. This facility allows activity diagrams to expand and show who has the responsibility for each activity in a process.
- We need to verify and validate the current set of functional models (use case, activity diagrams, use case descriptions) to ensure that they consistently represent the business processes under consideration.
- Walkthroughs is a manual approach that supports verifying and validating the evolving models.
- The purpose of a walkthrough is to thoroughly test the consistency of the functional models to the functional requirements.
- a walkthrough uncovers errors or faults in the evolving specification. However, a walkthrough does not correct errors but simply identifies them.