State and Activity Diagrams (UML)

UML

- Interaction diagrams for each use case
- Sequence and collaboration diagrams
- Sequence: Objects on top, line is object’s lifeline, arrow between 2 objects is a message, from top to bottom, conditions of message use [ ], iteration marker, return as dashed line.

UML

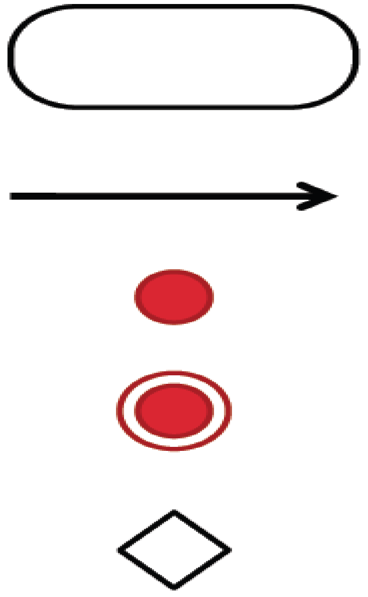
- COLLABORATION: objects as icons, arrows show messages of use case, sequence indicated by numbering messages, objectname:classname.
- Sequence emphasises sequence (order) while collaboration shows object connection, colaboration between object sin a use case.

State and Activity Diagrams

- State diagrams with superstates
- Concurrent state diagrams
- Shows behaviour of an object over several use cases.
- ACTIVITY DIAGRAMS:
- Flow of activities, triggers from activities with guards (logical true or false result), synchronisation bar (activities can occur in parallel order irrelevant)
- Can handle parallel processes

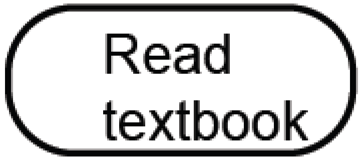
Modelling a Business Flow

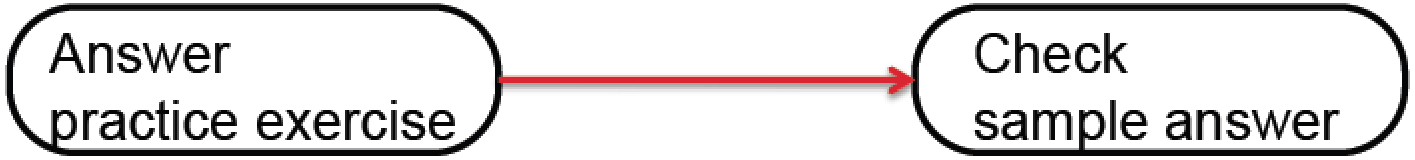
- What happens: activities
- When it happens: control flow
- Who/Where it happens: partitions
- How information is moved and transformed: object flow
  - Action or Activity
    - – Represents action or set of actions
  - Control Flow
    - – Shows sequence of execution
  - Initial Node
    - – The beginning of a set of actions
  - Final Node
    - – Stops all flows in an activity
  - Decision Node
    - - Represents a test condition
    - – Same symbol for merging afterwards, once common processing is resumed



Activities

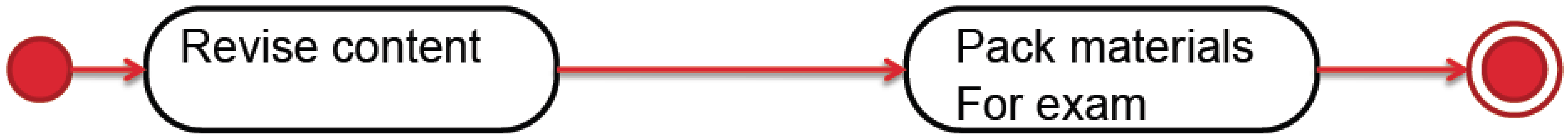
- An activity is behaviour that is performed for a purpose
- - It may be done by a person or by a computer-based system
- - An action is a special case, which won't be decompose behaviours
- Name activity with verb-noun
- - Eg "read textbook" or "do practice questions"
- Notation: rounded rectangle
- - Name placed inside
- Typically, in a business process, activities must happen in a particular
- sequence
- - One activity starts after another has finished
- - Eg "check solution against sample answers" should happen after "answer
- practice exercise"

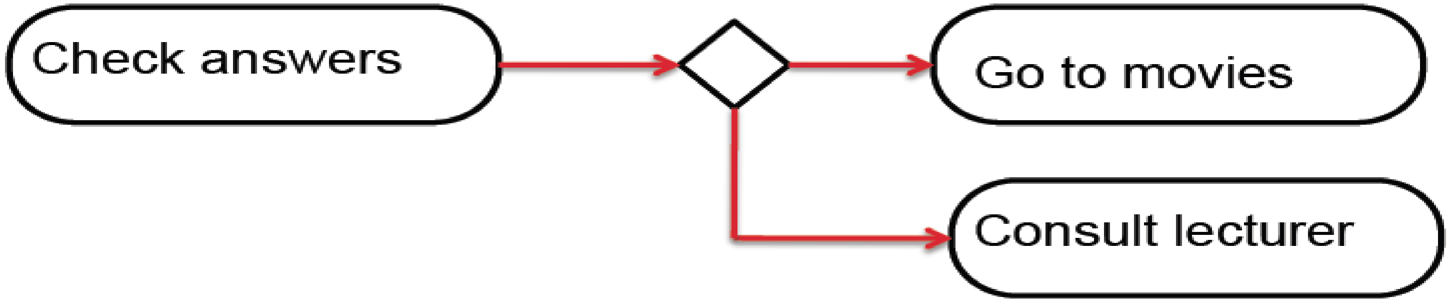


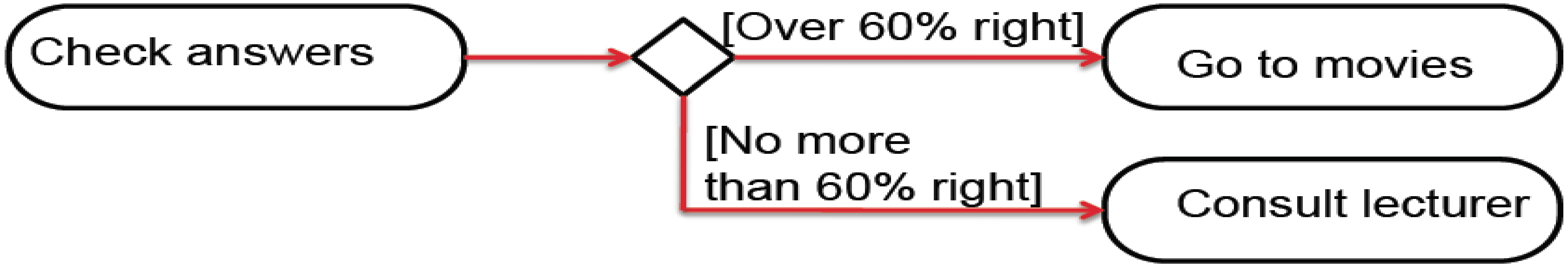


Modelling business process

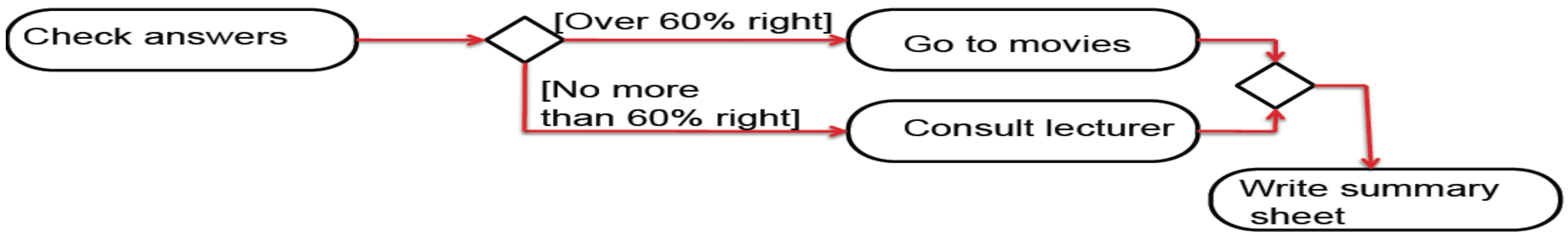
- Show an “initial node” to represent the start of the business process, and a “final node” for the finish
- Initial node is a filled-in circle
- - It has control flow arrow going to the first activity that happens
- There may be several final nodes
- Sometimes control goes along *one* among several different paths depending on circumstances
- An activity might take place, or it might not
- - Also, perhaps the difference is in the order of activities rather than which ones happen
- The making of a decision, that impacts on subsequent flow, must be shown
- Notation: Diamond, with one arrow incoming from the previous activity, and two or more outgoing, to the different activities that come next in different situations



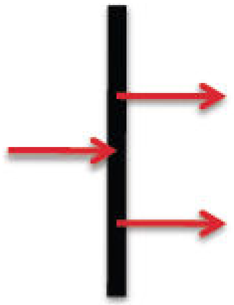


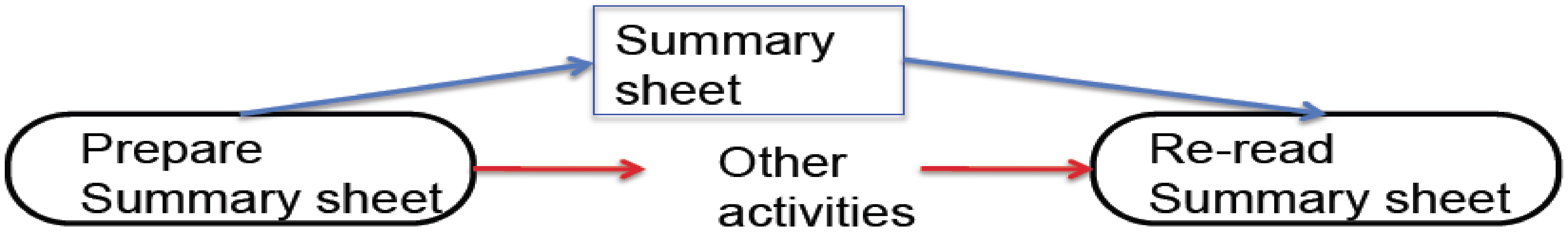


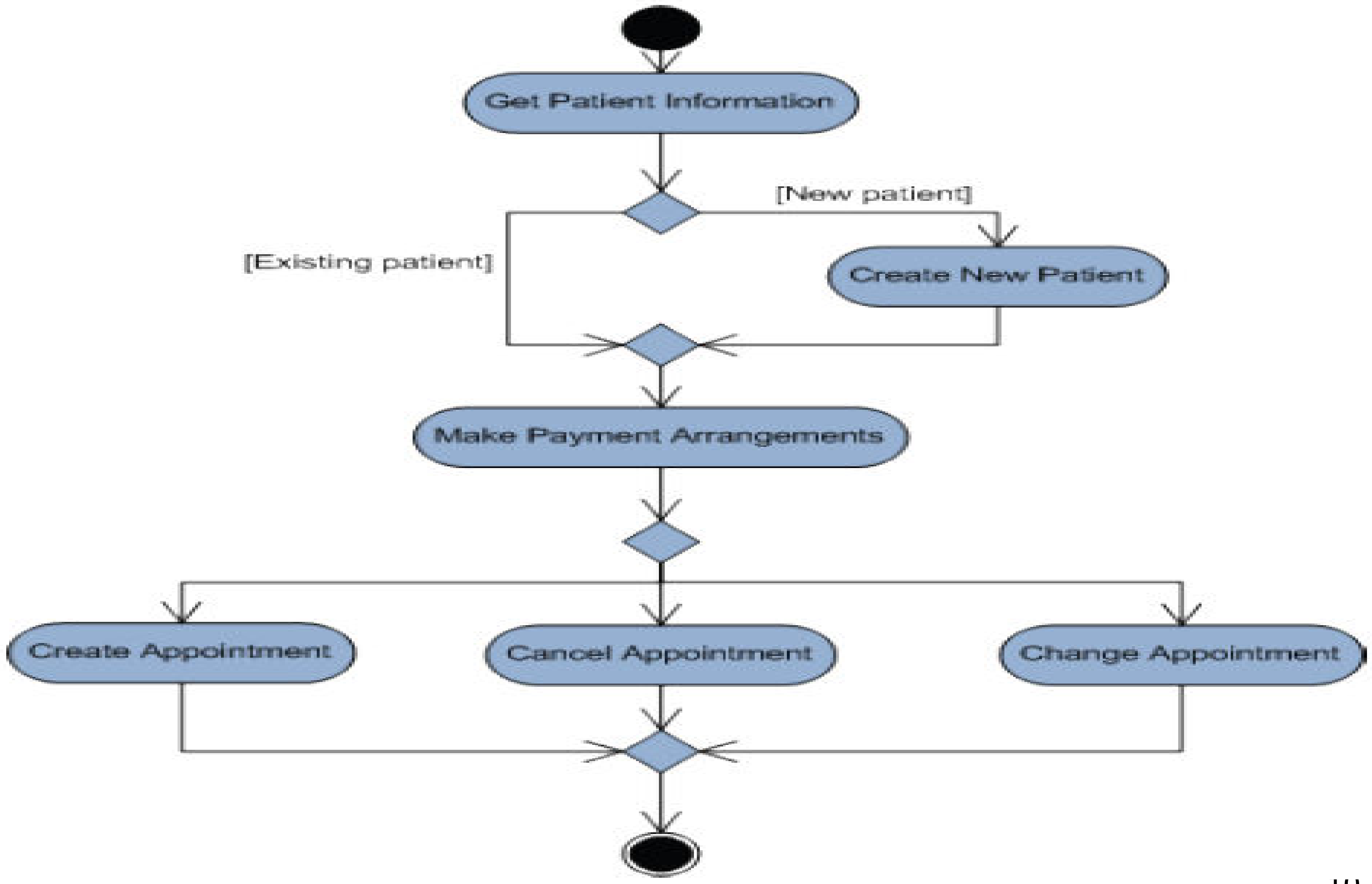
- It is wise to carefully document the situation that determines whether a particular branch is followed, after a decision node. This is a condition, of the system state, or the values being processed, or simply the result of the decision-making
- Notation: words that describe the situation, inside square brackets, placed on the control edge that leaves the decision node. You can use [else] for a branch, meaning “take this whenever no other branch is true”



- Usually, when a decision is made, after some processing that is different, we return to parts of the process that are not impacted by the choice made earlier
- A “merge node” is written to bring the different flows together. A diamond, with more than one incoming arrow, and one outgoing arrow
- Sometimes, there are activities which can be done in parallel. Often done by different people, but not necessarily so
- Control flow can divide or split, with all the following flows happening concurrently
- This means that there are no synchronization rules governing whether an activity in one path happens before or after or at the very same time, as an activity in another path
- Notation: solid line going at right angles to the arrows
- - Fork: One incoming arrow and several outgoing arrows
- - Join: Several incoming arrows, one outgoing arrow




Summary sheet
Prepare Summary sheet
Other activities
Re-read Summary sheet


Get Patient Information
[New patient]
[Existing patient]
Create New Patient
Make Payment Arrangements
Create Appointment
Cancel Appointment
Change Appointment

Class diagram

- Class diagrams with associations (has), generalisation (inherits), and aggregation (contains)
- Each class has a name, attributes, and operations.
- Aggregation (part of), composition (part to only one whole)

State Diagrams

- States an object and how state changes as events reach use case.
- Start → transition Event[Guard]/Action
- Eg. Item received[some items not in stock]/get next item
- →state with do/activity
- Eg. “Checking” do/check item