Introduction to UML

Slide Set - 7

Organized & Presented By:

Software Engineering Team, CSED

TIET, Patiala

Overview

- **What is Modeling?**
- **What is UML?**
- **Why UML?**
- **Understanding the basics of UML**
- **UML diagrams**
- **UML Modeling tools**

Modeling

- **Models are abstractions that portray the essentials of a complex problem or structure by filtering out nonessential details.**

- **Describing a system at a high level of abstraction**
  - **A model of the system**
  - **Used for requirements and specifications**

- **Models help us organize, visualize, understand, and create complex things.**

- **Is it necessary to model software systems?**

What is UML?

- **UML stands for “Unified Modeling Language”**
- **It is an industry-standard graphical language for specifying, visualizing, constructing, and documenting the artifacts of an object-oriented system under development.**
- **The UML uses mostly graphical notations to express the OO analysis and design of software projects.**
- **Simplifies the complex process of software design**

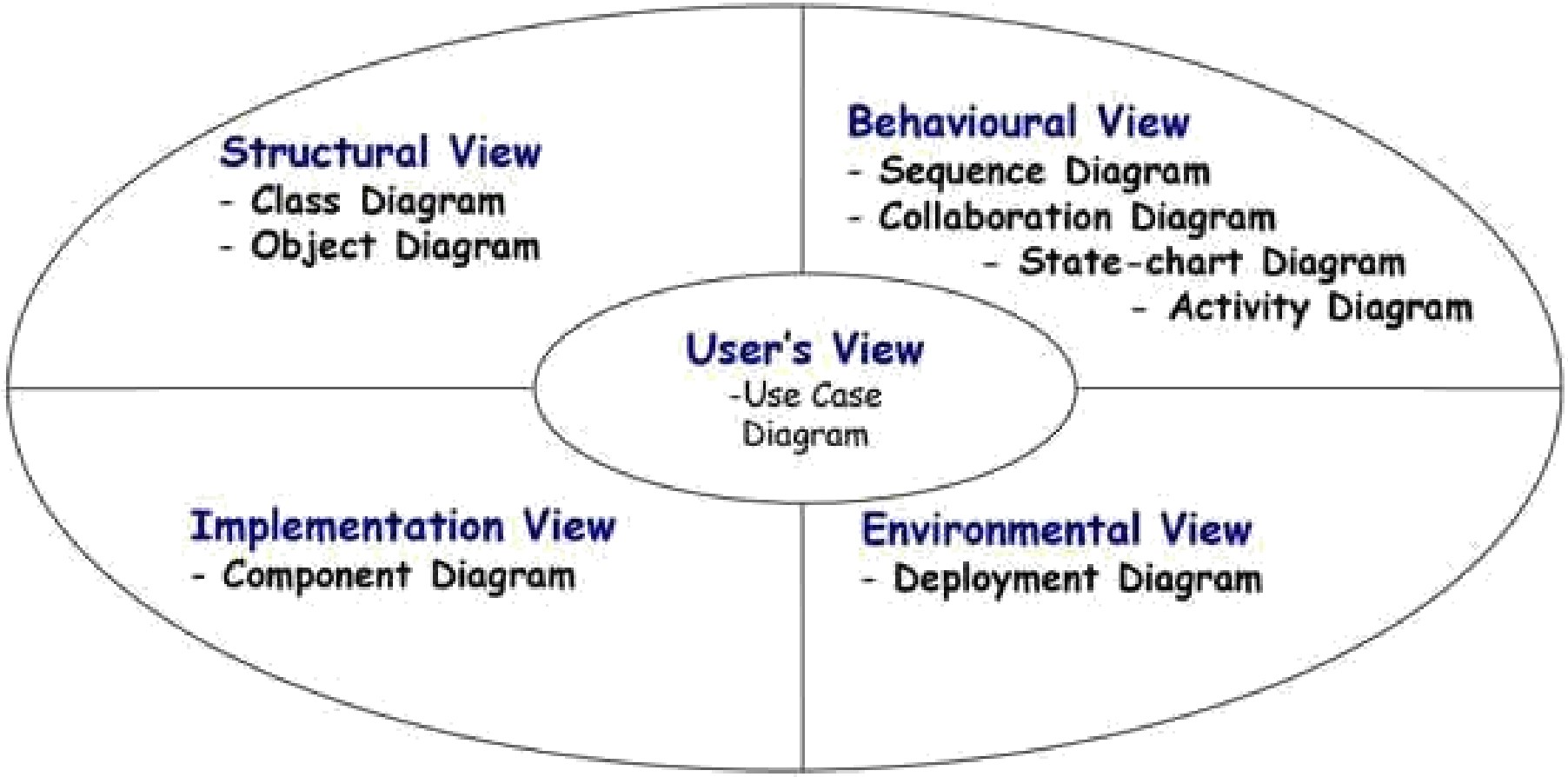
Why UML for Modeling

- **Use graphical notation to communicate more clearly than natural language (imprecise) and code(too detailed).**
- **Help acquire an overall view of a system.**
- **UML is not dependent on any one language or technology.**
- **UML moves us from fragmentation to standardization.**

History of the UML

- **In 1990s, many different methodologies, along with their own set of notations, were introduced to the market**
  - **– Object Modeling Technique (OMT) (James Rumbaugh)**
  - **– Grady Booch**
  - **– Object-oriented software engineering (OOSE) (Ivar Jacobson)**

UML Diagrams



Diagrams and views in UML

Source: slideplayer.com

UML Diagrams

- Use case diagram
- Activity diagram
- Class diagram
- Sequence diagram
- Collaboration diagram
- State diagram
- Object diagram
- Component diagram
- Deployment diagram

What is USE CASE diagram?

- **A use case diagram establish the capability of the system as a whole.**

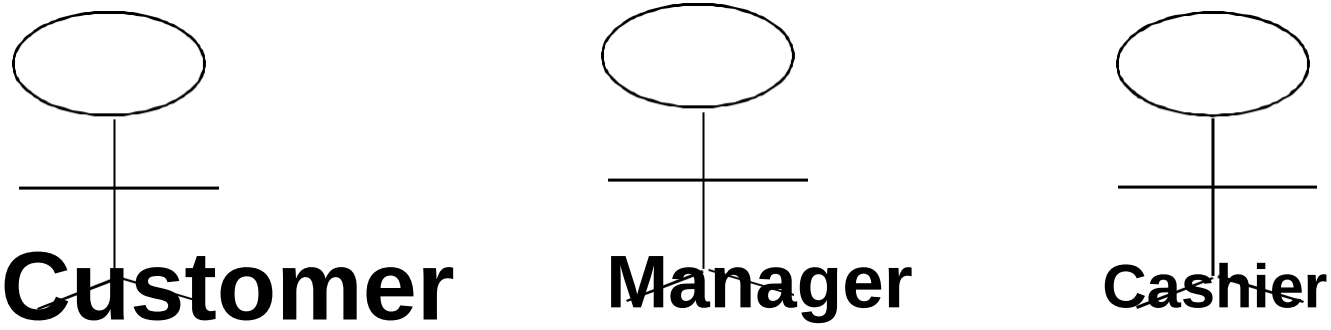
- **Components of use case diagram:**
  - **Actor**
  - **Use case**
  - **System boundary**
  - **Relationship**
  - **Actor relationship**

- **Semantic of the components is followed.**

ACTOR:

What is an actor?

- **An actor is some one or something that must interact with the system under development**
- **UML notation for actor is stickman, shown below.**



ACTOR:

More about an actor:

- **It is role a user plays with respect to system.**
- **Actors are not part of the system they represent anyone or anything that must interact with the system.**
- **Actors carry out use cases and a single actor may perform more than one use cases.**
- **Actors are determined by observing the direct uses of the system,**

ACTOR:

Contd…

- **Those are responsible for its use and maintain as well as other systems that interact with the developed system.**

- **An actor may**
  - **input information to the system.**
  - **receive information from the system.**
  - **input to and out from the system.**

ACTOR:

How do we find the actor?

- **Ask following questions to find the actors:**
  - **Who uses the system?**
  - **Who installs the system?**
  - **Who Starts up the system?**
  - **What other systems use this system?**
  - **Who gets the information from the system?**
  - **Who provides information to the system?**

- **Actor is always external to the system. They are never part of the**

  system to be developed.

ACTOR:

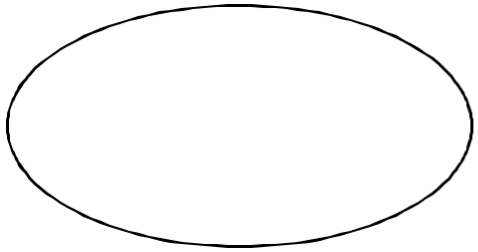
4-Categories of an actor:

- **Primary Actor : Who uses the main system functions.**
- **Secondary Actor: Who takes care of administration & maintenance.**
- **External h/w: The h/w devices which are part of application domain and must be used.**
- **Other system: The other system with which the system must interact.**

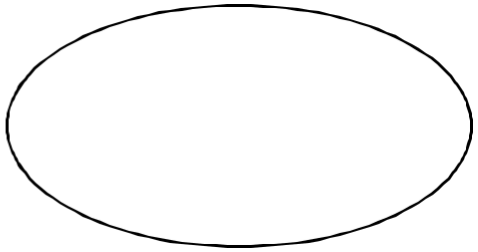
USE CASE:

What is USE case?

- **A use case is a pattern of behavior, the system exhibits**
- **Each use case is a sequence of related transactions performed by an actor and the system in dialogue.**
- **USE CASE is dialogue between an actor and the system.**
- **Examples:**



Open new account



Withdrawal of cash from ATM

SYSTEM BOUNDARY:

What is System Boundary?

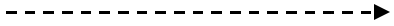
- **It is shown as a rectangle.**
- **It helps to identify what is external verses internal, and what the**

  responsibilities of the system are.
- **The external environment is represented only by actors.**

RELATIONSHIP between Use Cases

- **Relationship between two use cases is refereed as either include or extends.**

  <<includes>> Arrow is towards secondary use case

- **Multiple use cases share a piece of same functionality.**
- **This functionality is placed in a separate use case rather than**

  documenting in every use case that needs it.
- **<<includes>> relationship shows behavior that is common to one or more use cases**

RELATIONSHIP between Use Cases

<<extends>> Arrow is towards primary use case

- It is used to show optional behavior, which is required only

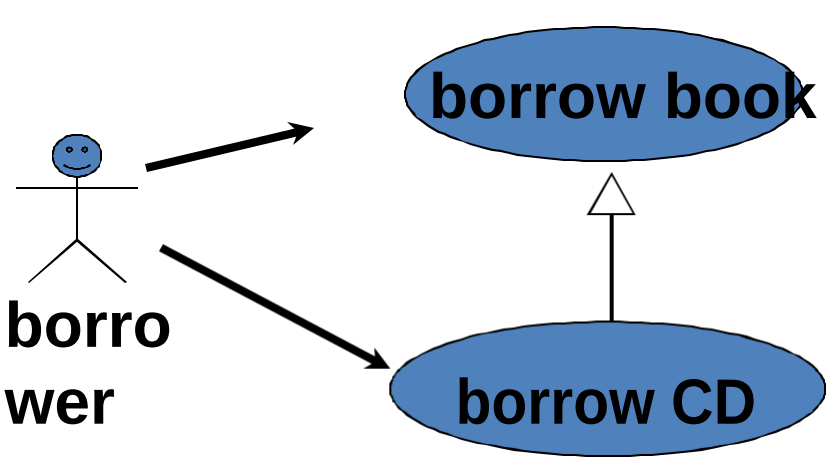
  under certain condition.

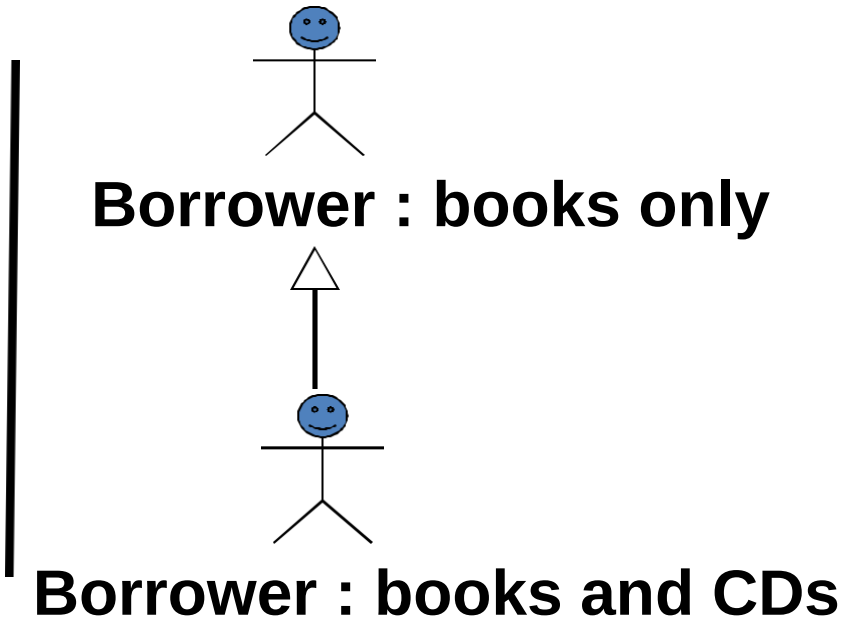
- e.g. Order and Rush Order: Rush Order is a special/optional case of order.

- Here Order is a normal order which is served within 30 minutes and user can select Rush Order instead, which means meal will be served within 10 minutes and user will have to pay Rs. 100 extra.

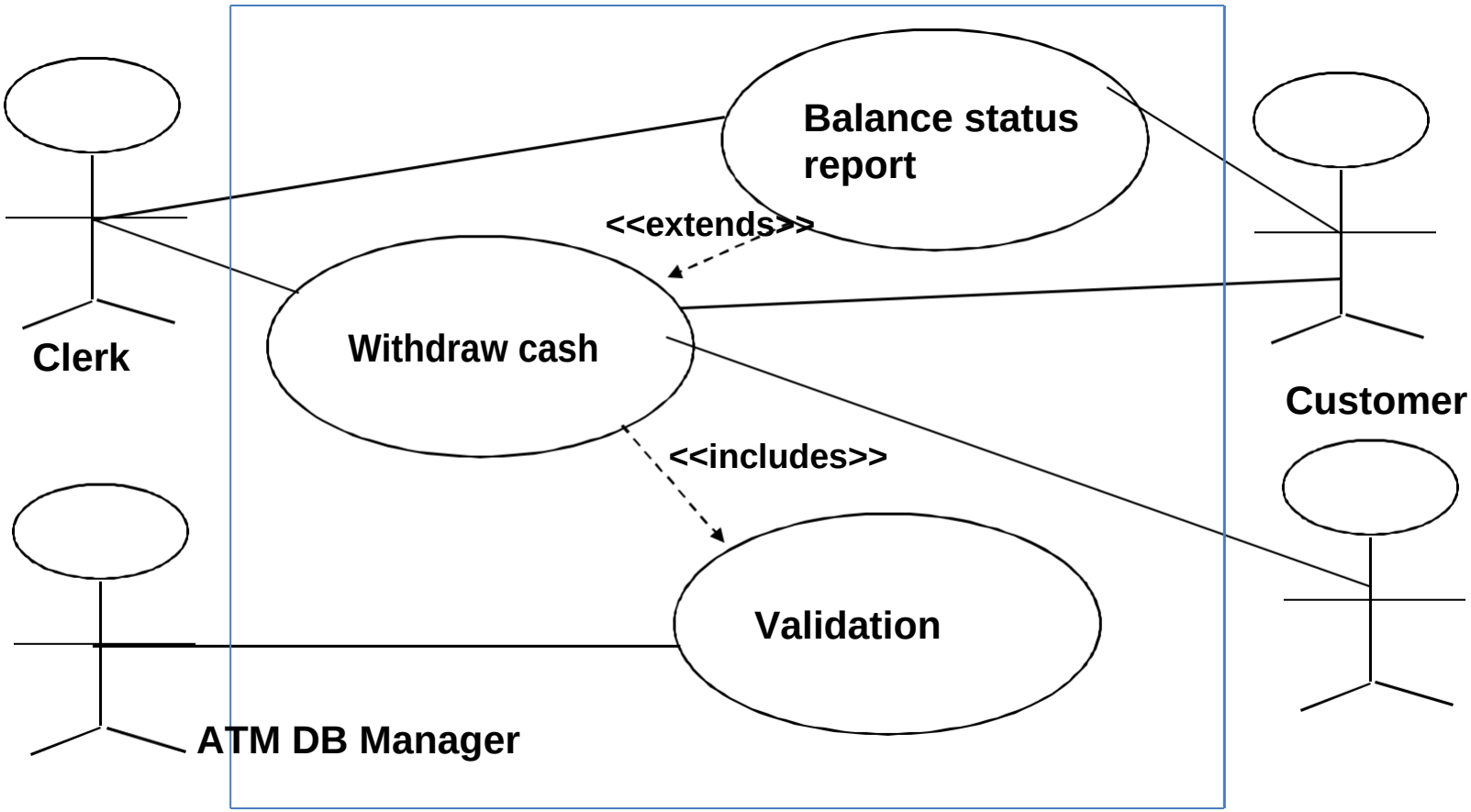
Generalisation

- Actors and use cases can be generalised, showing that one is simply a special case of the other
  - This is like inheritance in object-oriented design
  - Suppose that to borrow a CD, users must have additional registration and pay per CD.





USE CASE Diagram for Withdraw Cash



Typical Course of Events/ Successful Scenario/ Normal Scenario

(To be written in related Use Case Template)

For withdrawal of cash:

- **1.(SR) The ATM asks the user to insert a card.**
- **2.(AA) The user inserts a cash card.**
- **3.(SR) The ATM accepts the card and reads its serial number.**
- **4.(SR) The ATM requests the password.**
- **5.(AA) The user enters 1234.**
- **6.(SR) The ATM verifies the serial number and password with the**

  bank and gets the notification accordingly.
- **7.(SA)The ATM asks the user to select the kind of transaction.**
- **8.(AA)User selects the withdrawal.**

Normal Flow of Events:

- **9.(SR)The ATM asks for the amount of cash; user enters Rs. 2500/-**
- **10.(SR)The ATM verifies that the amount of cash is within predefined policy limits and asks the bank, to process the transaction which eventually confirms success and returns the new account balance.**
- **11.(SR) The ATM dispenses cash and asks the user to take it.**
- **12.(AA) The user takes the cash.**
- **13.(SR) The ATM asks whether the user wants to continue.**
- **14.(AA) The user indicates no.**
- **15.(SR) The ATM prints a receipt, ejects the card and asks the user to take them**
- **16.(AA) The user takes the receipt and the card.**
- **17.(SR) The ATM asks a user to insert a card.**

Alternative Flow of Events:

For withdrawal of cash use case:

- **9. The ATM asks for the amount of cash; the user has change of mind and hits the “cancel”.**

Exceptional Flow of Events:

For withdrawal of cash use case:

- **3 Suspicious pattern of usage on the card.**
- **10 The machine is out of cash.**
- **11 Money gets stuck in the machine.**

USE CASE Template

- **Generic format for documenting the use case:**

- Pre condition: If any

– Use case: Name of the case.

– Actors :List of actors(external agents), indicating who initiates the use case.

– Purpose : Intention of the use case.

– Overview : Description.

– Type : primary / secondary.

USE CASE Template Contd.

- **Typical Course of Events/ Successful Scenario/ Normal Scenario**

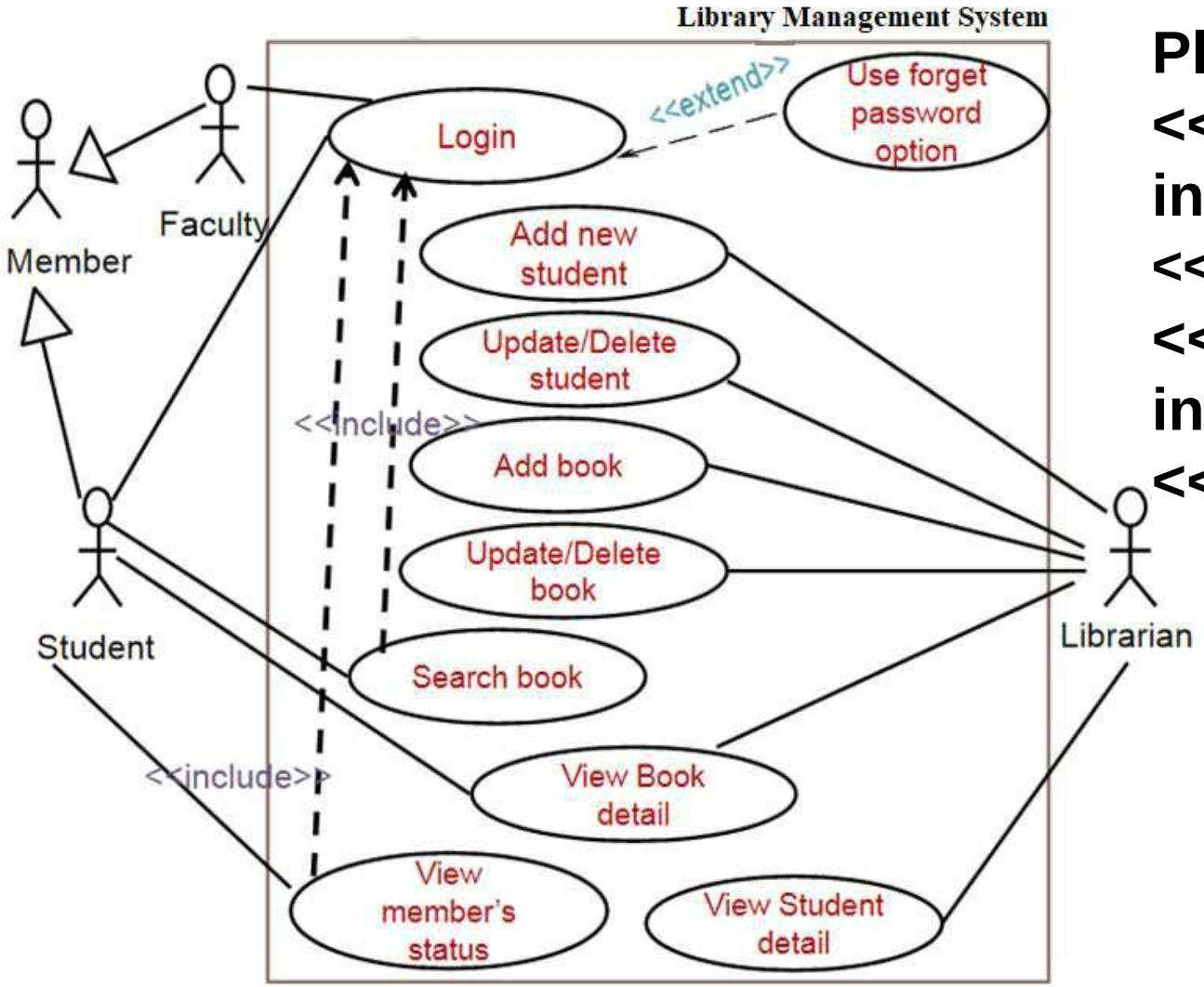
  ACTOR ACTION(AA) : Numbered actions of the actor.

  SYSTEM RESPONSE(SR): Numbered description of system responses.

  SYSTEM ACTION(SA): Numbered description of system responses.

- **Exceptional Flow of Events/ Extension points**
- **Alternate Flow**
- **Post Condition(if any)**

Another Example – Library Management System



Pls Note: Use <<extends>> instead of <<extend>> and <<includes>> instead of <<include>>

Sample Use Case Template– Library Management System

1. Use Case Title	Search Book
2. Abbreviated Title	Search Book
3. Use Case Id	5
4. Actors	Member
5. Description With this search facility, user can specify any of search criteria. For example, title, author, publication, accession no. etc., and search the desired book.	
5.1. Pre Conditions: User must be logged in(session)	
5.2. Task Sequence 1. Search screen will be shown by the system 2. Select search criteria and enter the required information. 3. On clicking the search button, system will show search results.	
5.3. Post Conditions: 1. User can view his desired results. 2. User can go for another search.	
6. Modification History: Date 21-dec-2018	
7. Author: James Henry	

Activity diagrams

- Activity diagrams are
  - a technique to describe procedural logic, business process & work flow
  - similar to flowcharts, but the latter does not support parallel behaviour
  - used to model a specific use case at a more detailed level
- States
  - have transitions to other states
  - may be either
    - action states: atomic action
    - activity states: actions that can be broken down
  - may have decisions, explained by conditions & merges
  - may have forks & joins for modelling simultaneous activities

Activity diagram notation

- **Action/Activity**
  - **– Long rounded rectangle**

- **Start/Initial State**
  - **– unique with only one transition line**

- **End State**
  - **– not necessarily unique**

- **Transition**
  - **– Lines indicate sequential flow of actions. Must be labeled after a decision.**

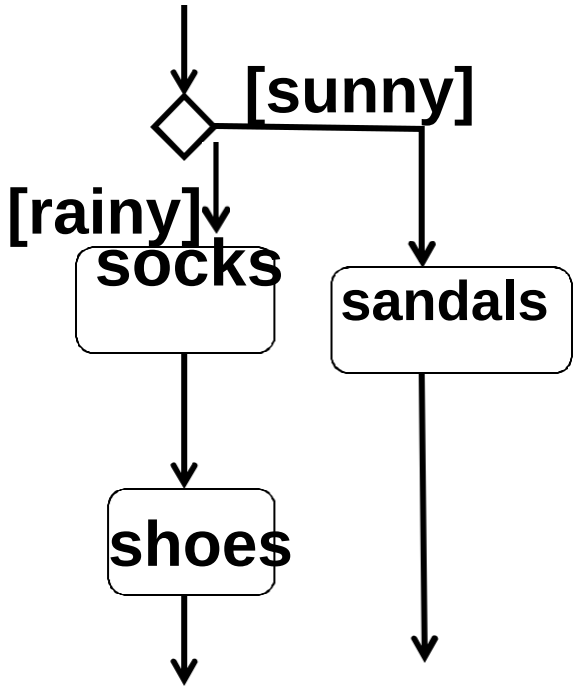
  - **– Arrows indicate direction of next action**
- **Activity is shown in more detail in another diagram**

- **Condition**

Decision Points & Guards

NOTE: Please stick to notation of activity While drawing diagrams

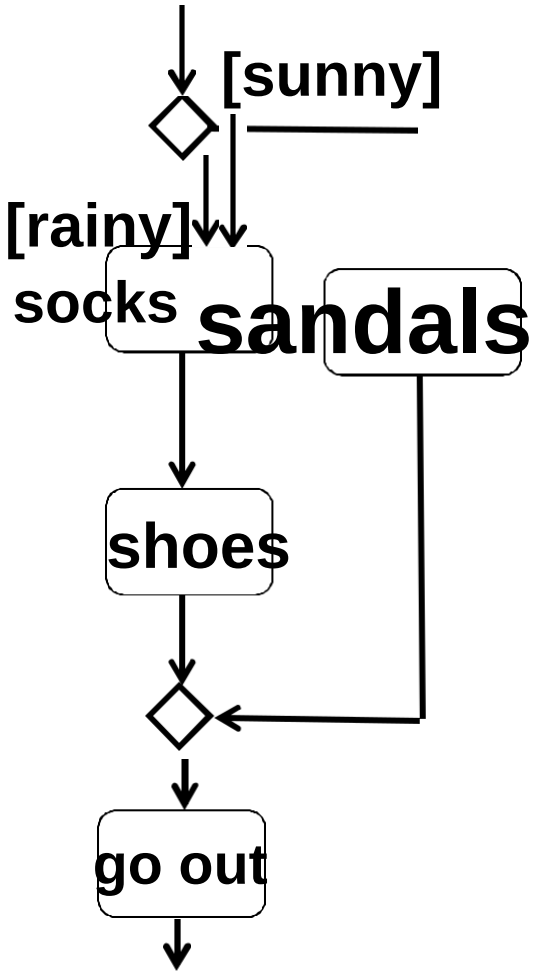
- **Decision point : diamond**
  - **– Single entry transition line**
  - **– Multiple exit transition lines**
    - **each with Guard (condition)**
    - **exit lines must cover all eventualities**

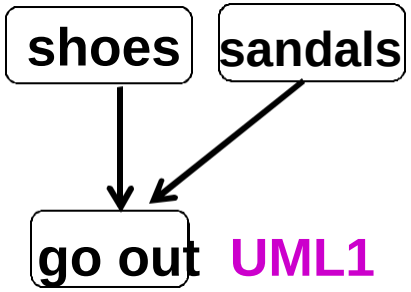


- **There could be a problem with the decisions in this diagram. What do you think it is?**

Merge Points

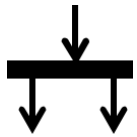
- **Merge : diamond**
  - **– Are used to indicate a merge after a decision**
  - **– Multiple entry, single exit**
  - **– In UML1, more than one transition line canenter an activity.**
  - **– In UML2 (and in this module), use merge diamonds.**



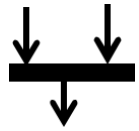


Synchronisation

- Sometimes certain action sequences can be done in parallel (synchronously, concurrently, simultaneously)
- Fork



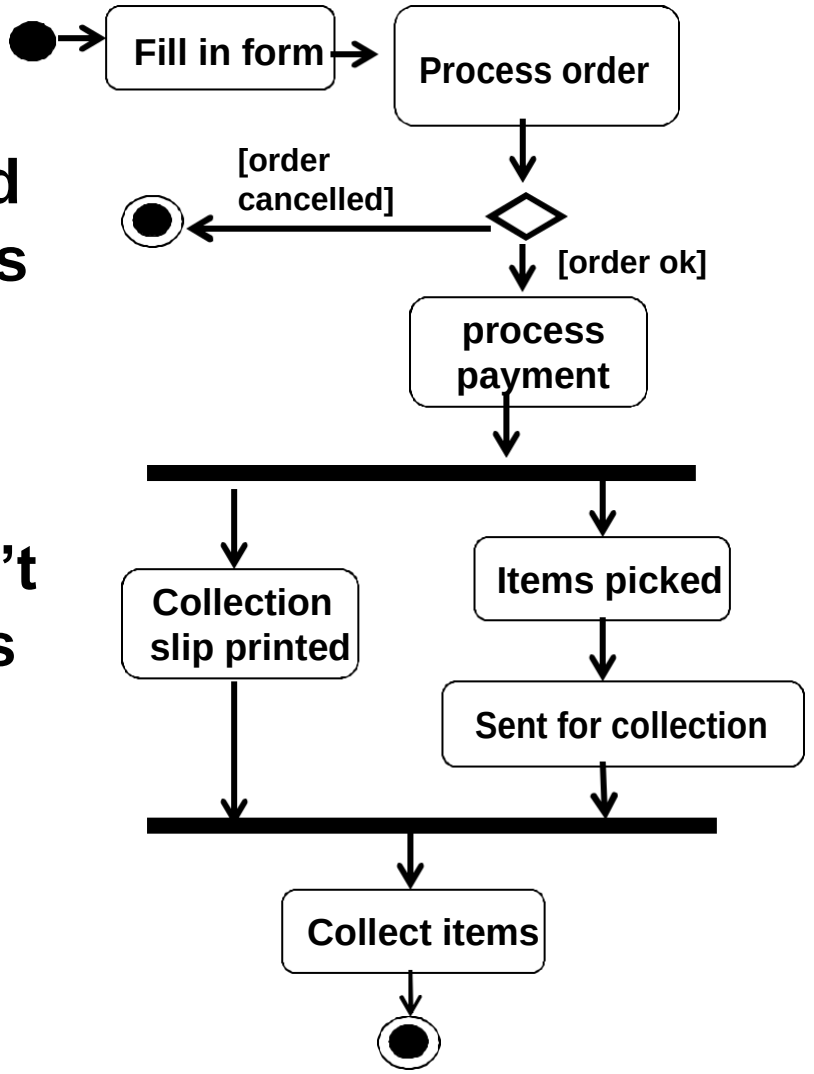
  - Shows start of concurrent actions
  - The thread within each exit can be executed at the same time as the other thread
  - Single entry, multiple exits
- Join
  - Shows end of concurrent actions
  - The exit cannot occur until all entry threads have been completed
  - Multiple entry, single exit



Example: Online Shopping

-The fork shows that the collection slip can be printed at the same time as the items are being picked out and sent for collection.

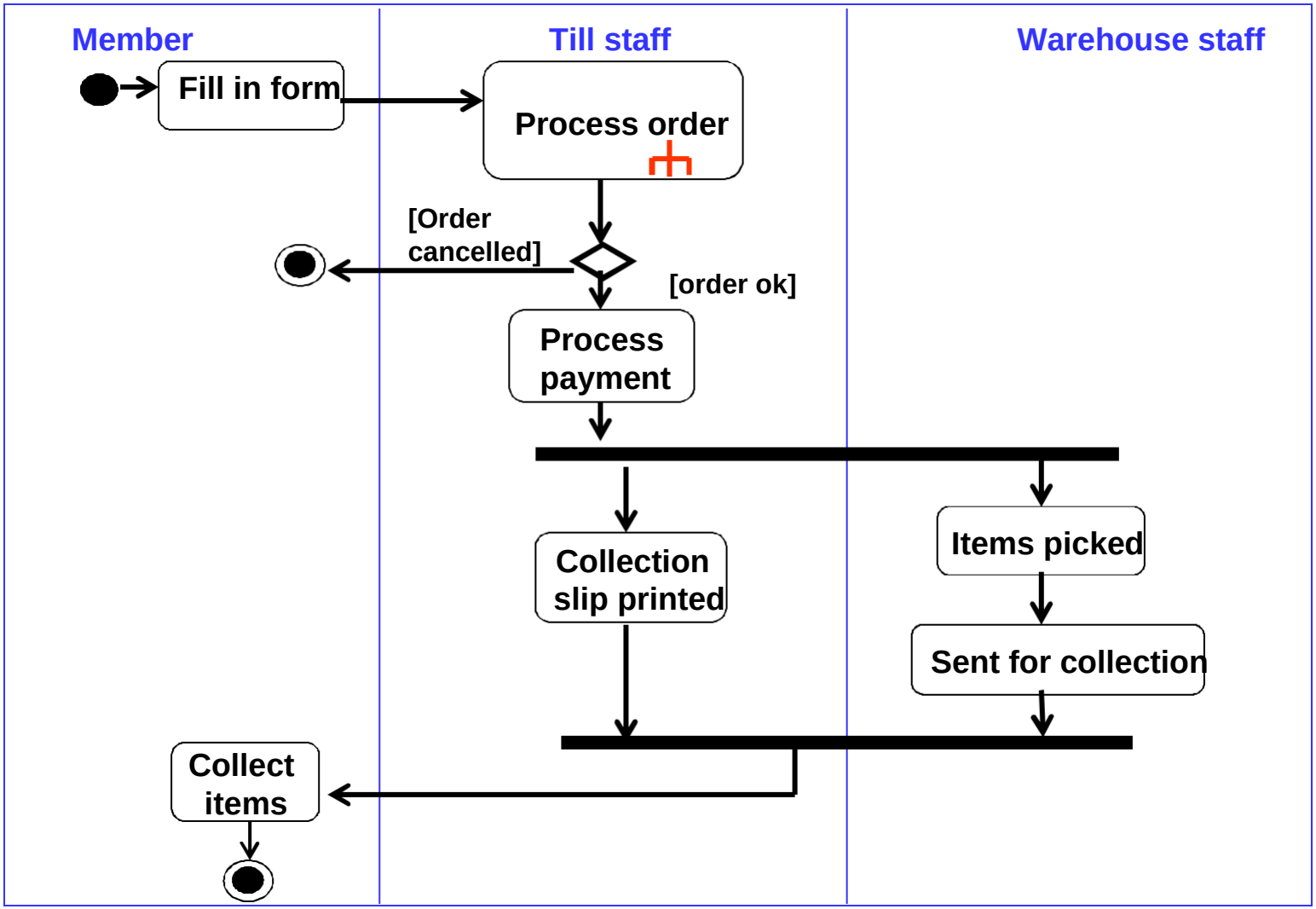
-The join shows that you can't collect them until the slip has been printed and also the items have been picked out and sent for collection.



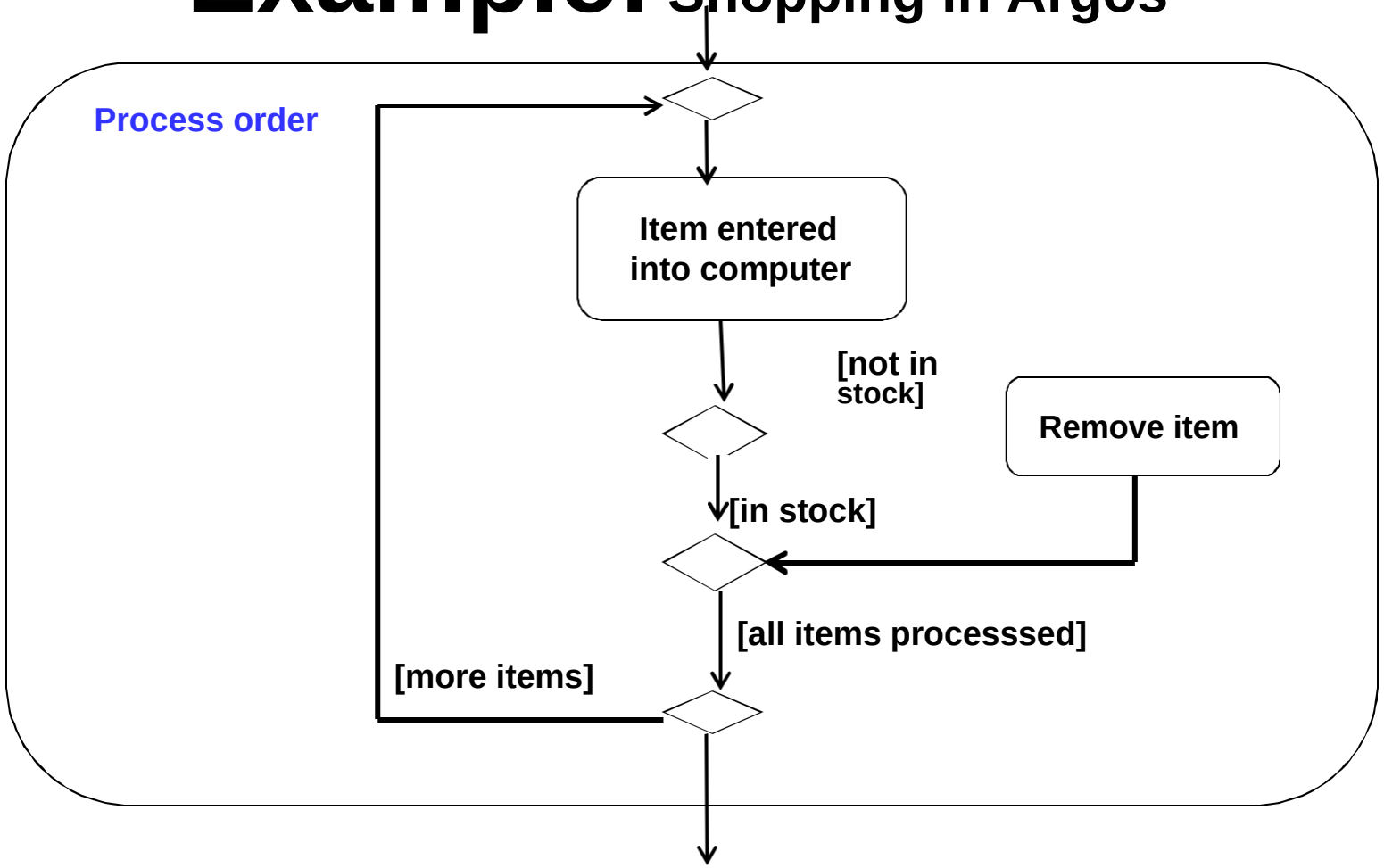
Swimlanes / partitions

- **Useful to model activity's procedural flow of control between objects**
- **Use vertical columns**
- **Place object’s name at the top of the column**
- **Place each action associated with an object in that object's swimlane**

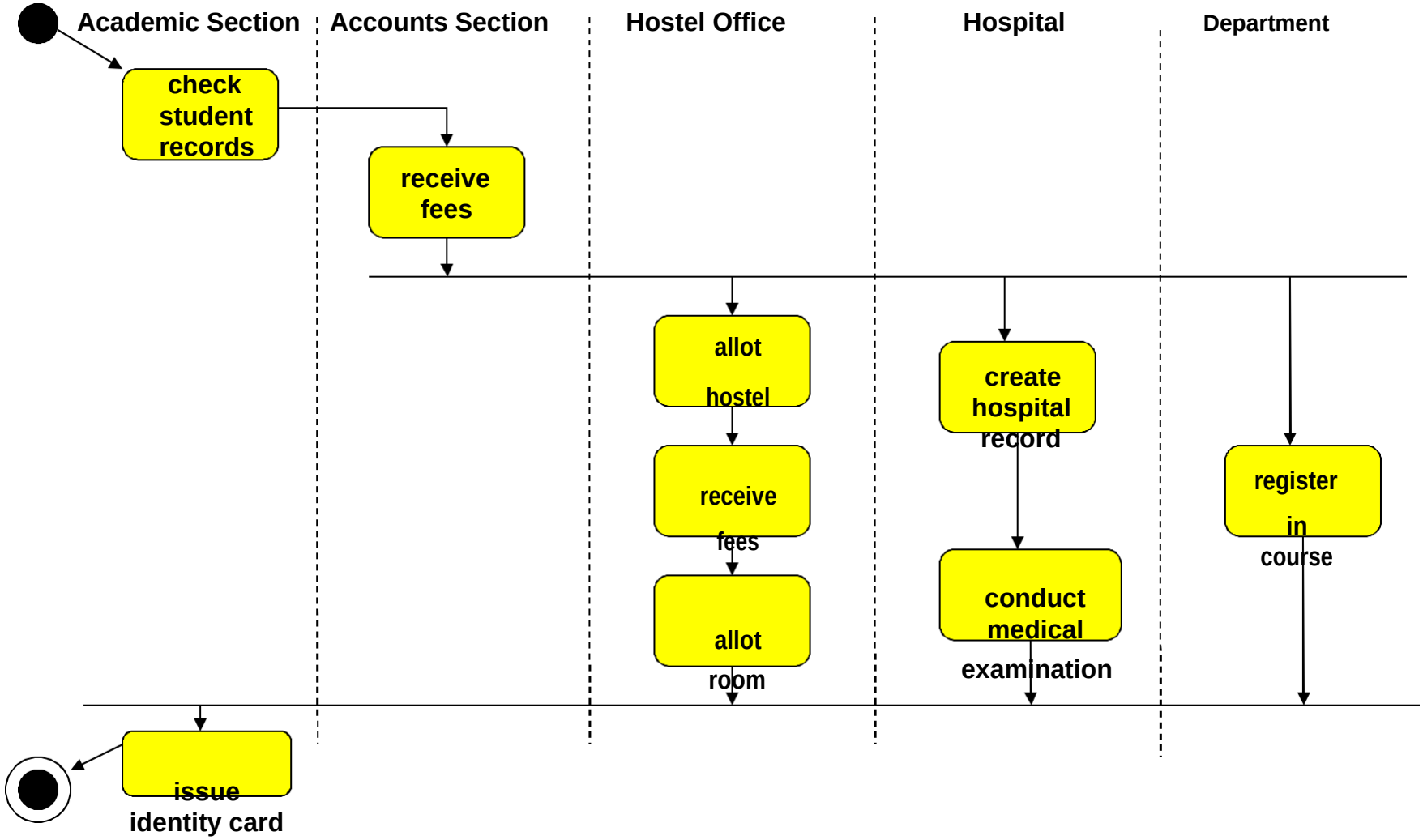
Example: Online Shopping



Example: Shopping in Argos



Another Example of an Activity Diagram



Activity diagram for student admission procedure