Module 42: Software Engineering

UML - Use Case Diagrams

Intructors: Abir Das and Sourangshu Bhattacharya

Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

{*abir, sourangshu*}*@cse.iitkgp.ac.in*

Slides taken from NPTEL course on Object-Oriented Analysis & Design

by **Prof. Partha Pratim Das**

Module Objectives

- Understanding Use Case Diagrams

Module Outline

- Use Case Diagrams
- Actors
- Use Cases
  - Specification of Use-Cases

What are Use case Diagrams

- The integration of business knowledge with the development specification is a requirement
- The development organization knows the specifications for developing a module, but does not know who will interact with these modules and for what purpose
- Use Case Diagrams depicts the human interaction with the system to give the context of who uses, which part of the system and for what purpose

Components of Use case Diagrams

The use case diagram is composed of

- Actors
- Use cases and their specifications
- Relationships between Use cases

Actor

- Actors are entities that interface with the system
- They can be people or other systems
- Actors are depicted as stylized stick figures
- This stick figures are used as stereotypes to depict many models at the same time labeled with guillemets <<>>
- The roles the actors play in the system is important, not their real world identity

Actor

Actors can be classified as

- **human**: e.g. novice/trained user; system administrator
- **non-human**: e.g., fax, e-mail
- **primary**: ultimate user of the system
- **secondary**: ensures the correct functionality of the system
- **active**: initiates use cases
- **passive**: corresponding use case is initiated by the system

Actor

How can Actors be identified?

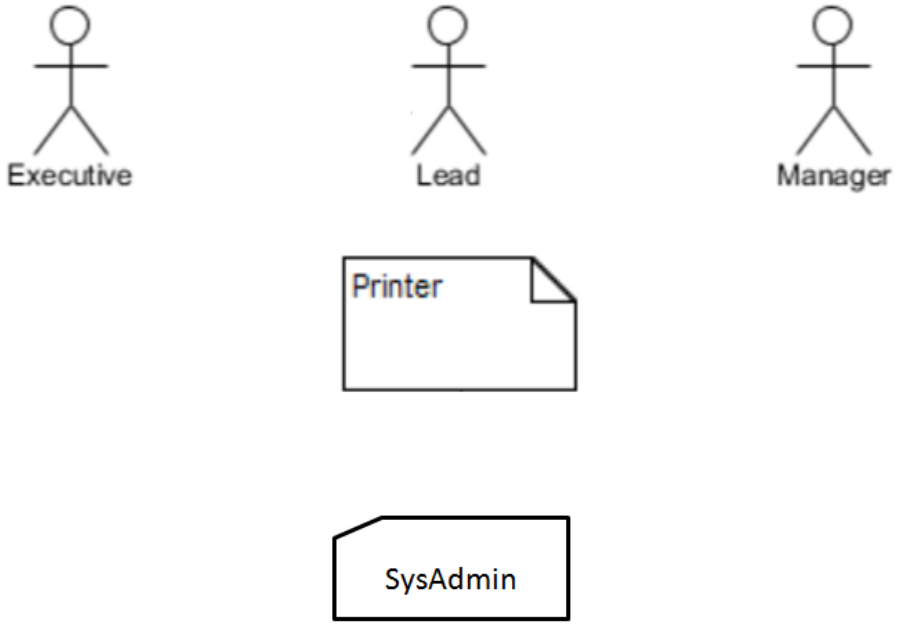
- Who uses the essential use cases?
- Who needs system support in order to fulfill the daily tasks?
- Who is responsible for system administration?
- What are the external devices/software systems the system has to communicate with?
- Who is interested in the results of the system?

LMS: List of all Identified Nouns: RECAP (Module 17)

Company	Attendance	Leave	Employees
Contributors	Lead	Executive	Manager
Leave Rules	Days	Year	Name
Type of Leave	Period	Absence	Holiday
PL	CL	EL	DL
SL	ML	LWP	UL
Pre-approval	Month	Service	Quarter
Medical Certificate	Parenthood Certificate	Disciplinary Action	Administration Function
Daily Attendance	Personal Details	Calender Year	Batch Task
Account	Balance	Designation	SysAdmin
Parent	Salary	Week	List
Privilege	Right	Login ID	Leave Status
Employee Code			

Actors in LMS

- In the LMS, the human actors are Manager, Lead and Executive.
- The non human actors is the printer
- A Secondary actor is the SysAdmin



Use Cases

- Use cases represent what the actors want your system to do for them
- Each use case is a complete course of events to be executed in the system from a user's perspective
- Use cases can contain short descriptions course of events in the system from a user's perspective

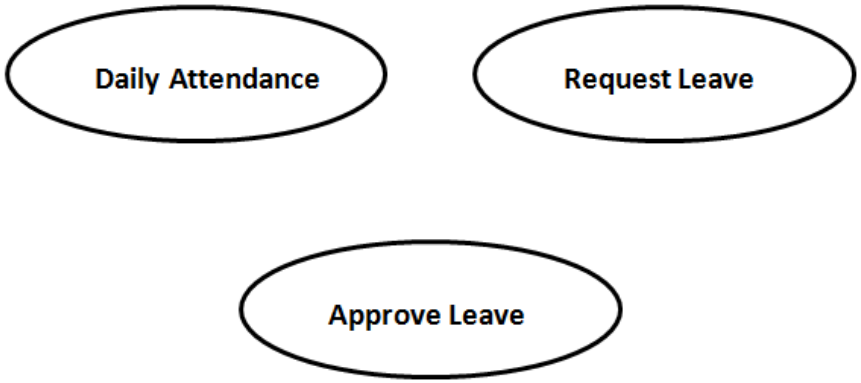
LMS: List of all Identified Verbs: RECAP (Module 18)

Wants	Manage	Work	Report
Approve	Regret	Credit	Join
Prorate	Cross	En-cash	Paid
Allow	Send	Need	Become
Enjoy	Avail	Proceeding	Employ
Consider	Deduct	Provide	Request
Cancel	Check	Export	Revoke
Debit	Adjust	Perform	Hire
Fire	Generate	Leave En-cashment	Can be Availed
Can be Clubbed	Can't be Availed	Can't be Carried forward	Can't be Clubbed
Can't be Continued	Accumulated Up	Proposed for	Join Back
Doesn't Draw	Can be Revoked	Leave Credited	

Many extracted verbs are in derived forms – so we extract the unique stems

Use Cases in LMS

- Shown below are important use cases executed by actors of Leave Management System. Request Leave, Daily Attendance are use cases, that is, functionality which the all the three human actors need from the system. Similarly approve leave use case executed by only Lead and Manager



Specification of Use Case

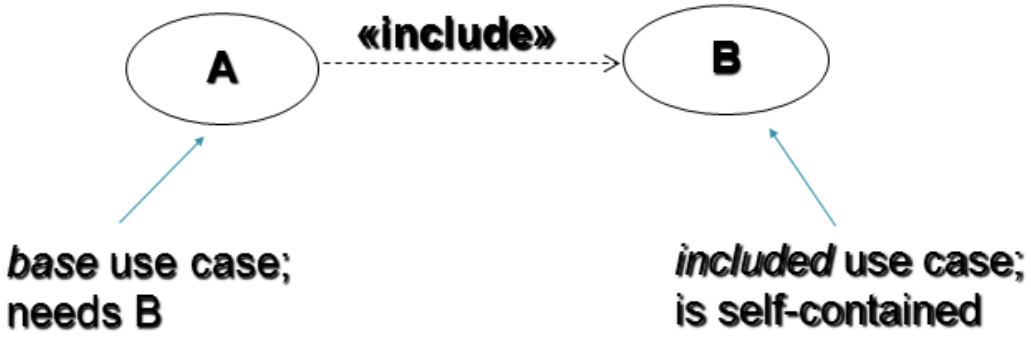
- A use case can be specified in the following manner. Depicted below for Request Leave.
  - **Use Case Name:** Request Leave
  - **Use Case Purpose:** The Executive, Lead and Manager, all of them uses this functionality to request for leaves in the system.
  - **Use Case Pre-condition:** User has login id and password to enter the system
  - **Use Case Post-condition:** A new leave request is entered in the system for the user
  - **Failure Conditions:** User does not have valid credentials to enter the system, the user do not have sufficient leave balance
  - **Actors:** Lead, Executive, Manager
  - **Optimistic Flow:**
    - ▷ The Executive, Lead and Manager checks for the leave balance.
    - ▷ If leave balances are available, leave request added.

Relationships among Use-Cases

- Use-Cases share various kinds of relationships
- A relationship between two Use-Cases is basically a dependency between the two Use-Cases
- Defining the relationship between two Use-Cases is the decision of the modeler of the use case diagram
- We discuss the following three relationships among Use-Cases
  - <<include>>
  - <<extend>>
  - Generalization

Relationship among Use-Cases: <<include>>

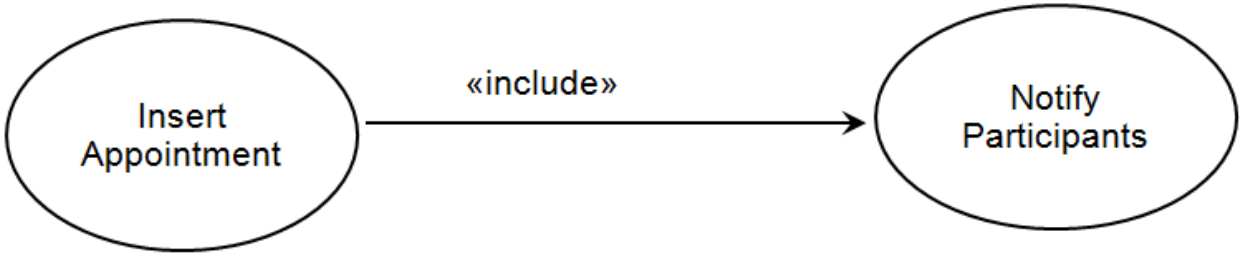
- The <<**includes**>> relationship involves one Use-Case including the behavior of another Use-Case in its sequence of events and actions
- Thus, the includes relationship explores the issue of reuse by factoring out the commonality across Use-Cases
- <<**includes**>> relationship:



  - the behavior of B is included into A
  - the included use case B is necessary to ensure the functionality of the base use case A

Relationship among Use-Cases: <<include>>

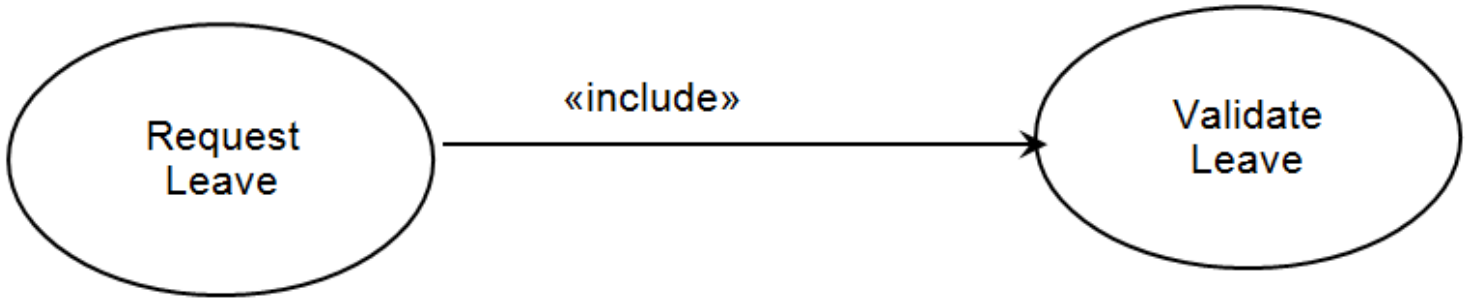
- In a calendar, if an appointment is inserted by *Insert Appointment*, the participants will be notified of appointment by included Use-Case *Notify Participants*
- The included Use-Case *Notify Participants* is necessary to ensure the functionality of the base Use-Case *Insert Appointment*



- The Use-Case *Notify Participants* may be included in other Use-Cases – for example, in *Cancel Appointment*

<<include>> in LMS

- *Validate Leave* Use-Case is included in the Use-Case *Request Leave*
- The include Use-Cases *Validate Leave* is necessary to complete the functionality of the base *Request Leave*



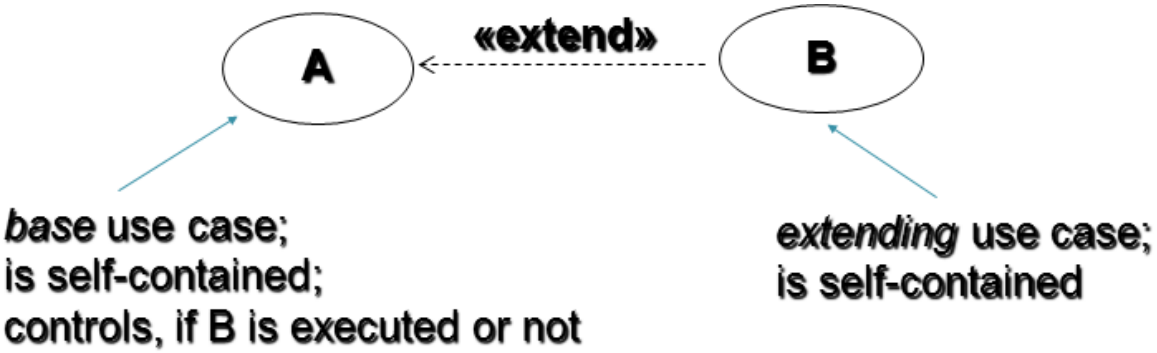
- The Use-Case *Validate Leave* will be included in *Approve Leave* too – and so on

Relationship among Use-Cases: <<extend>>

- The <<**extend**>> relationship among the Use-Cases is used to show optional system behavior
- An optional system behavior is extended only under certain conditions, known as **Extension Points**

Relationship among Use-Cases: <<extend>>

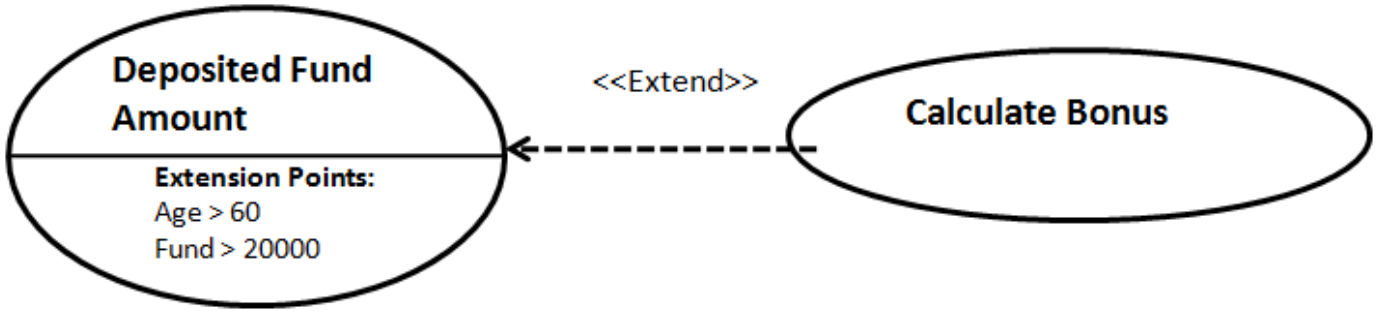
- <<**extend**>> relationship:



  - the behavior of B may be incorporated into A
  - the extending Use-Case B may be (but need not be) activated by the base Use-Case A
  - **extension points** specify the location where the extending Use-Cases extends the base Use-Case
  - the **condition** under which the extending Use-Case is incorporated has to be specified
  - more than one extension point can be specified for each Use-Case
  - the names of extension points have to be unique
  - the names of extension points need not be equal with the names of the extending Use-Cases

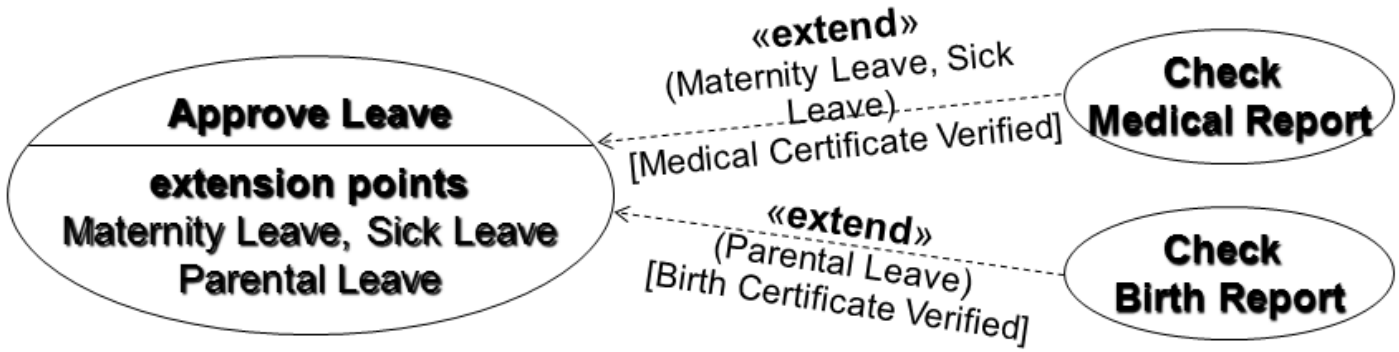
Relationship among Use-Cases: <<extend>>

- In a savings bank account, bonus is provided only if the deposited fund is above 20,000 or the depositor is above the age of 60 years
- The behavior of *Calculate Bonus* may be incorporated into *Deposited Fund amount*
- At **extension point**: Age above 60 years, Deposit $>$ 20,000



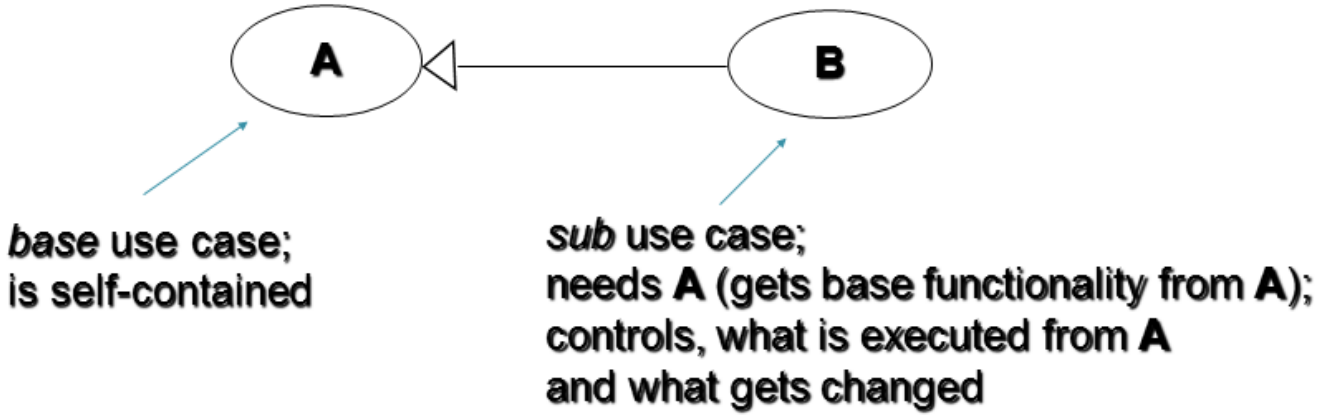
<<extend>> in LMS

- The behavior of *Check Report* may be incorporated into *Approve Leave*
- The extending Use-Case may be (but need not be) activated by the base Use-Case *Approve Leave* at extension points: Medical Leave, Maternity Leave
- Extending Use-Case (*Check Report*) extends the base Use-Case (*Approve Leave*)



Relationship among Use-Cases: Generalization – Concept of Hierarchy

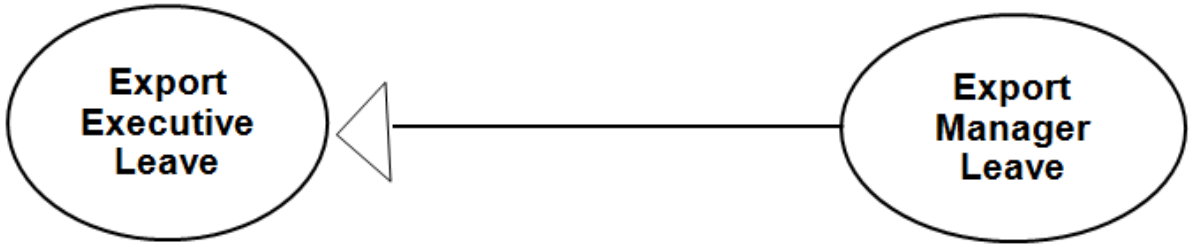
- Generalization works the same way with Use-Cases as it does with classes
- The child Use-Case inherits the behavior and meaning of the parent Use-Case
- Generalization helps us to depict the hierarchy present between Use-Cases
- Generalization relationship:



  - Similar to the generalization relationship between classes
  - B **inherits** the behavior of A and is allowed to **override** and **extend** it
  - B **inherits** all relationships of A
  - Modeling of **abstract Use-Cases** is also possible (abstract)

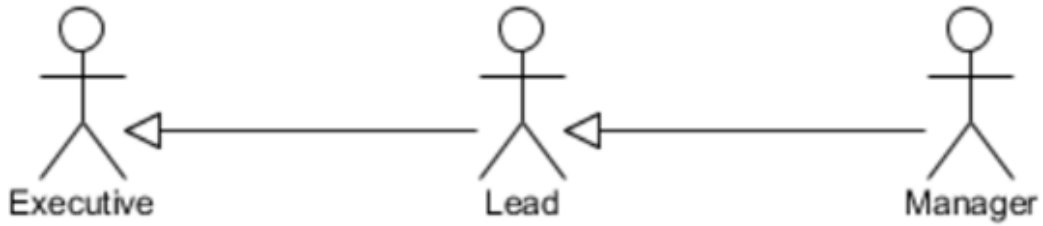
Generalization in LMS

- In LMS, *Export Manager Leave* Use-Case inherits the behavior of the parent Use-Case *Export Executive Leave*, that is, it contains the list of leaves but along with it, it contains additional work responsibilities of each of the executives. Hence a specialized Use-Case



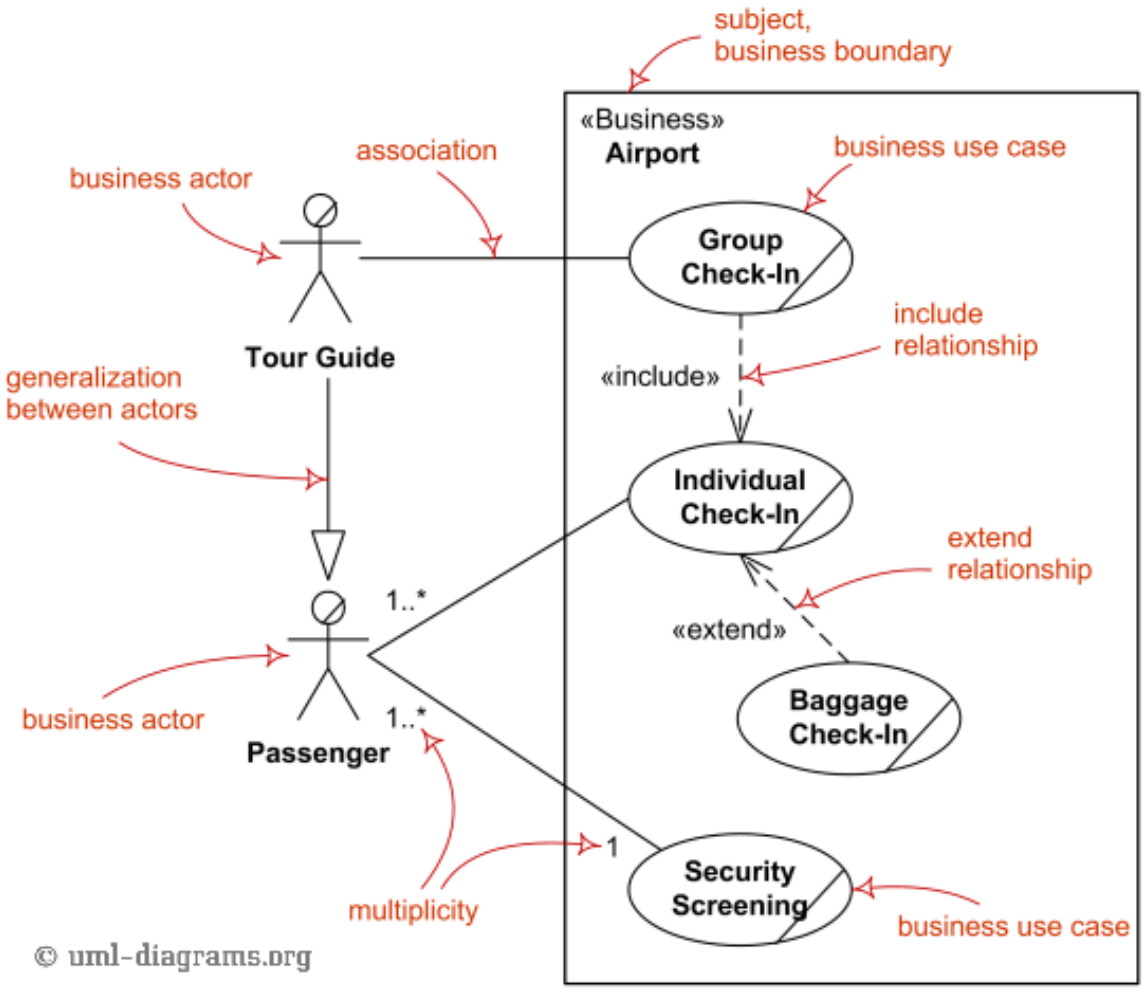
Generalization among Actors

- There can be generalization among actors, which can be captured in the use case diagrams
- Generalization among actors, specify that use cases executed by the base actor is inherited by the derived actor
- The derived actor can execute extra use cases



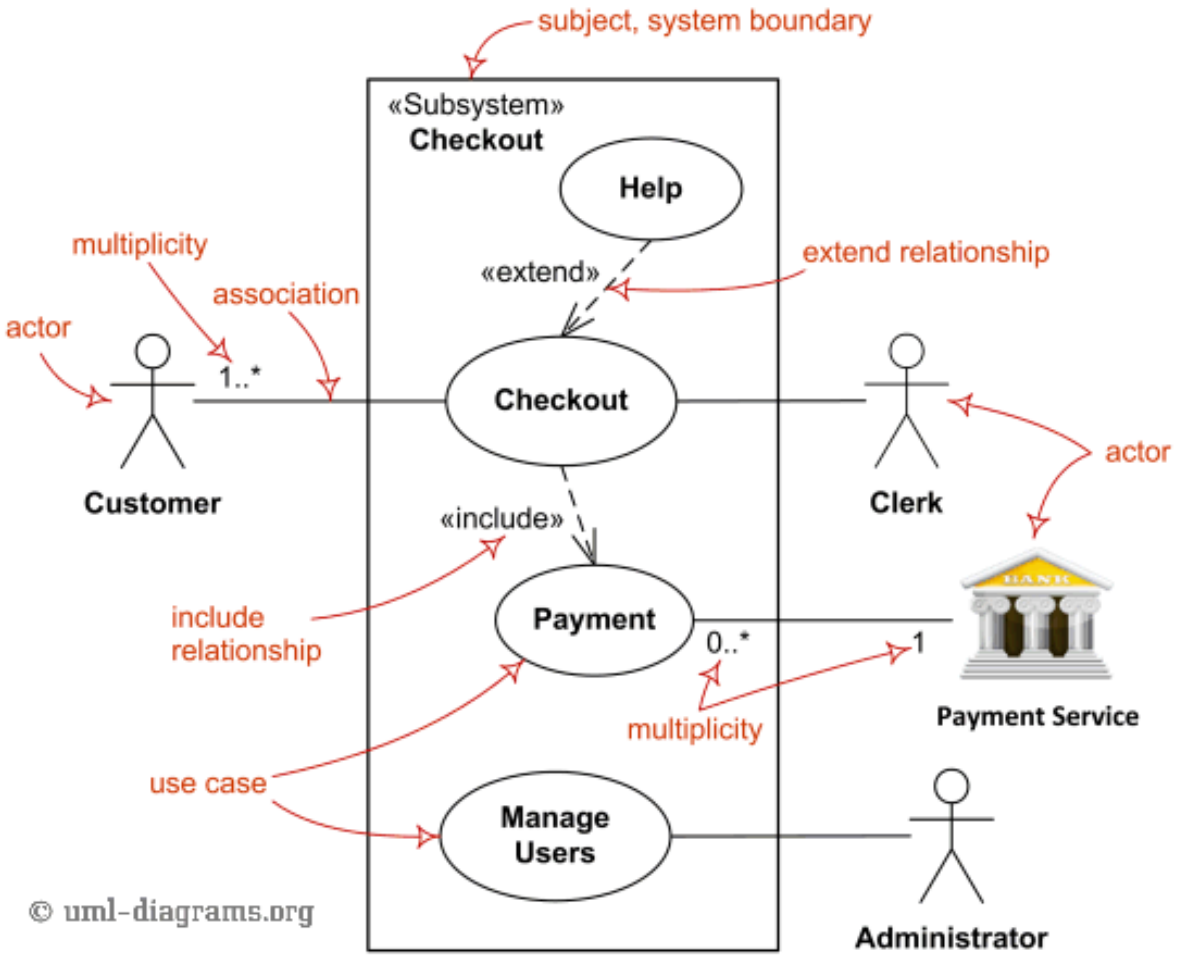
Generalization Among Actors

Use-Cases Example: RECAP (Module 24)



Source: *UML 2.5 Diagrams Overview*: http://www.uml-diagrams.org/uml-25-diagrams.html (10-Aug-16)

Use-Cases Example: RECAP (Module 24)



Source: *UML 2.5 Diagrams Overview*: http://www.uml-diagrams.org/uml-25-diagrams.html (10-Aug-16)

Identify Actors

Identify Actors for LMS

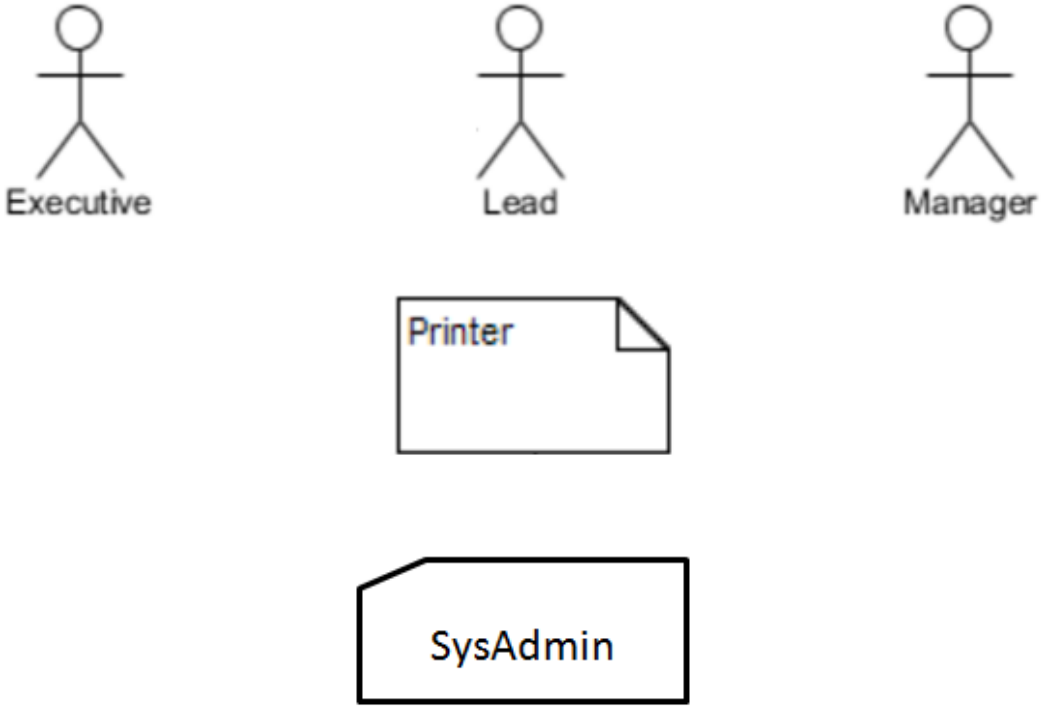
List of all Identified Nouns: RECAP (Module 17)

Company	Attendance	Leave	Employees
Contributors	Lead	Executive	Manager
Leave Rules	Days	Year	Name
Type of Leave	Period	Absence	Holiday
PL	CL	EL	DL
SL	ML	LWP	UL
Pre-approval	Month	Service	Quarter
Medical Certificate	Parenthood Certificate	Disciplinary Action	Administration Function
Daily Attendance	Personal Details	Calender Year	Batch Task
Account	Balance	Designation	SysAdmin
Parent	Salary	Week	List
Privilege	Right	Login ID	Leave Status
Employee Code			

Actors are Nouns

- Reading through the specification of the Leave Management System, we identity the actors
- Actors are the nouns in the specifications
  - Executive
  - Lead
  - Manager
  - Printer
- Now based on people or things, we identify the human actors and non-human actors
  - Human
    - Executive
    - Lead
    - Manager
  - Non-Human
    - Printer
  - Secondary
    - SysAdmin

Actors in LMS



Identify Use-Cases

Identify Use-Cases for LMS

List of all Identified Verbs (Stem Only): RECAP (Module 18)

Wants	Manage	Work	Report
Approve	Regret	Credit	Join
Prorate	Cross	En-cash	Pay
Allow	Send	Need	Become
Enjoy	Avail	Proceed	Employ
Consider	Deduct	Provide	Request
Cancel	Check	Export	Revoke
Debit	Adjust	Perform	Hire
Fire	Generate	Club	Carry forward
Continue	Accumulate	Propose	Join Back
Draw			

Use-Cases are Verbs

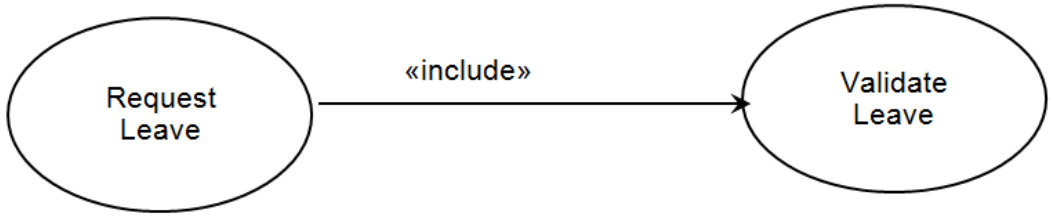
- Reading through the specification, we identify several verbs which are behaviors, that is, Use-Cases of the system
  - Daily Attendance
  - Request Leave
  - Cancel Leave
  - Avail Leave
  - Export Leave
  - Approve Leave
  - Revoke Leave
  - Export Exec Leave
  - Check Medical Report
  - Adjust
  - Debit
  - Credit

Identify Relationships

Identify Relationships for LMS

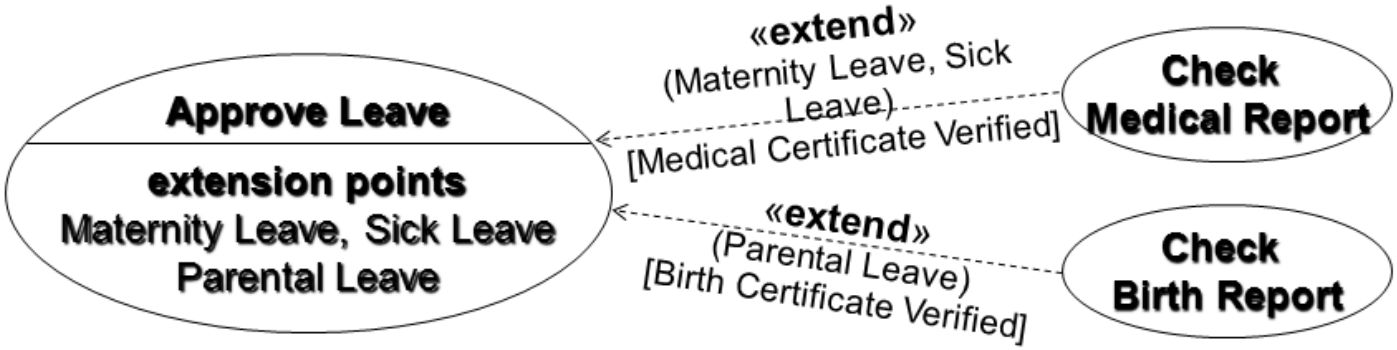
Includes

- Some verbs will have auxiliaries, that is, related tasks, required to complete its action
- Like *Request Leave* will include the verb *Check Leave*. Hence an includes definition is identified



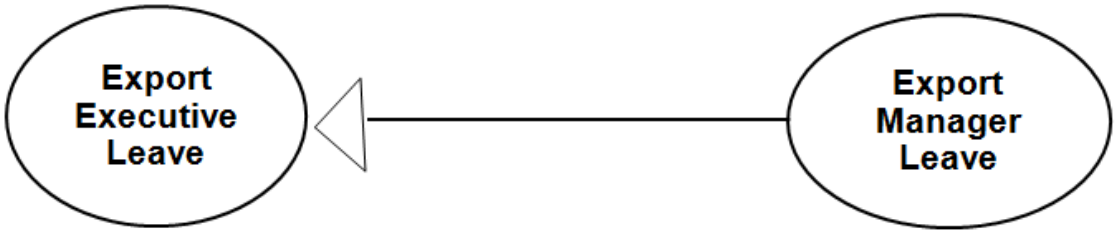
Extend

- Some verb will incorporate another verb at some conditions, that is, extension points
- In case of Medical or Maternity Leaves, the verb *Approve Leave* will be extended by the action *Check Medical Report*



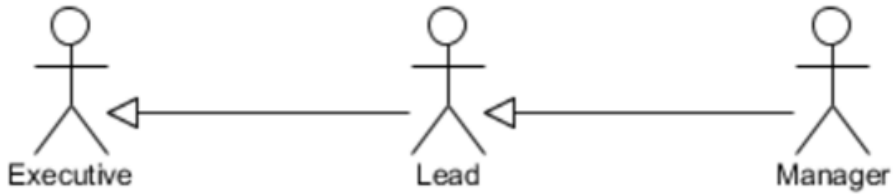
Generalization among Use-Cases

- There can be hierarchy among Use-Cases, that is, one of the Use-Cases inherits the behavior of another Use-Case, along with some extra behaviors
- We see *Export Exec Leave* has the same behavior like *Export Leave* (will be executed by all actors), along with some special actions for the Lead and Manager, as only they will be executing the Use-Case *Export Exec Leave*



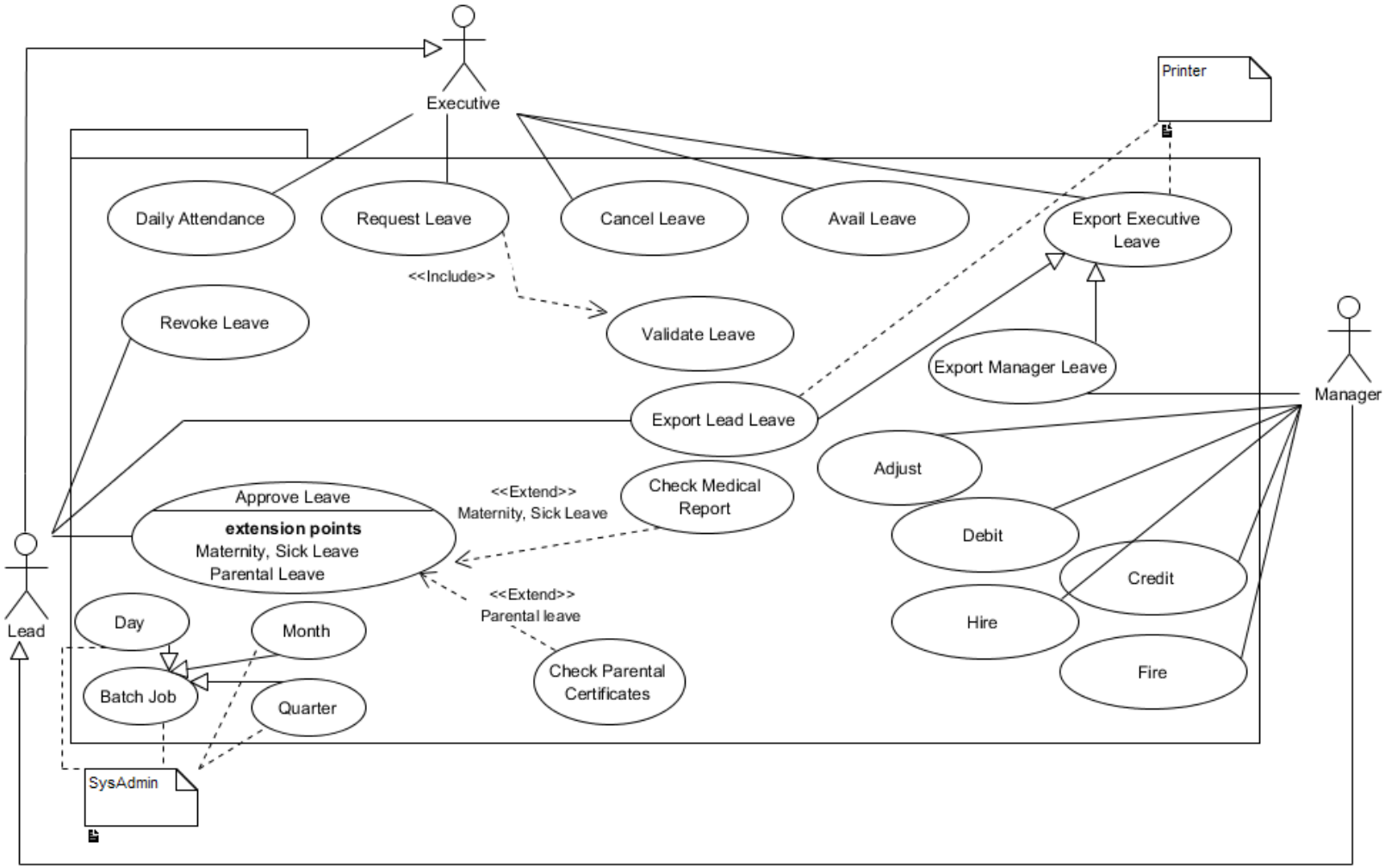
Generalization among Actors

- There can be hierarchy among the actors, that is, nouns.
- That is the base actor will execute some use cases, and the specialized actors, will execute extra Use-Cases
- Lead is an Executive, Manager is a Lead – means that
  - Lead executes all the use cases executed by an Executive along with some additional use cases (Revoke Leave, Approve Leave)
  - Similarly, Manager executes all the Use-Cases executed by a Lead along with some additional Use-Cases (Adjust Leave, Credit Leave)



Generalization Among Actors

Use-Case Diagram: Leave Management System



Not all use cases are shown in details

Module Summary

- Use case diagrams help to integrate business knowledge with the development specification
- Use Case diagram is used to model the various behaviors (usecases) of a system, and the external elements using and executing them (actors)
- Relationships among Use-Cases are classified and discussed
- Relationships among Use-Cases are identified for LMS
- Illustrated Use-Case diagrams for LMS