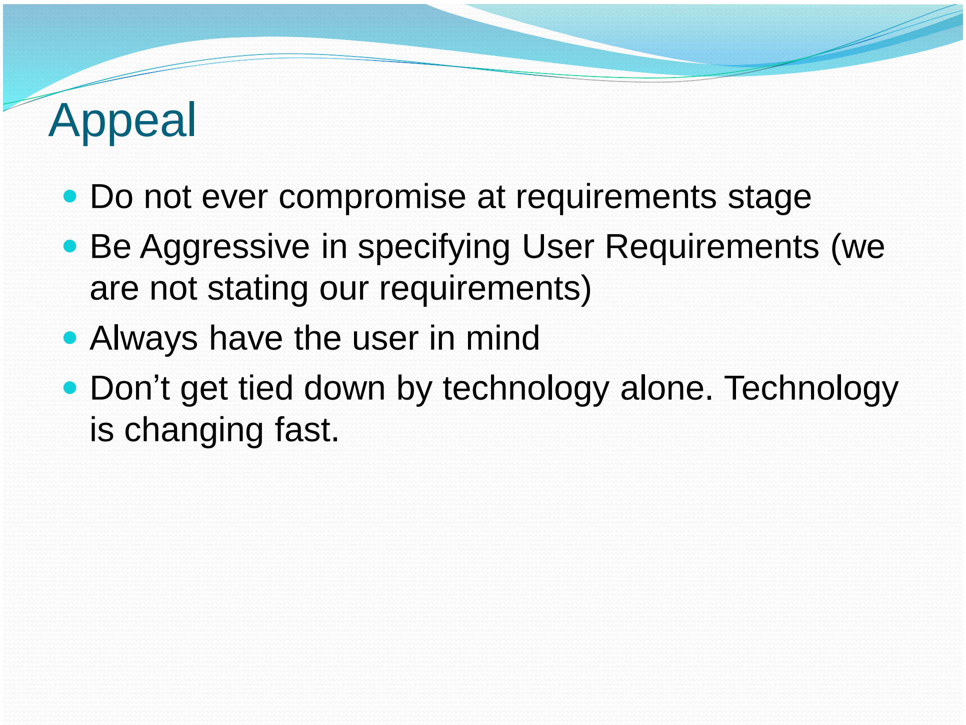
Data Modeling

Use case Diagram for Requirement specification

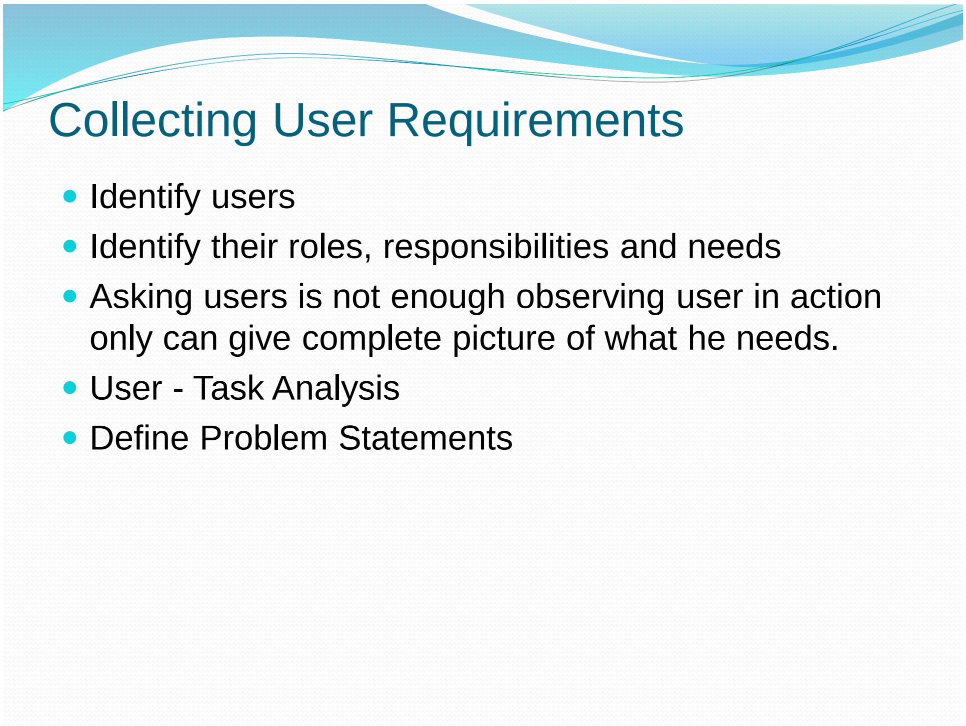
What can go wrong in a product: examples

- Rich in features
- Poor in presentation
- Interface Not intuitively designed
- Usability issues



Appeal

- Do not ever compromise at requirements stage
- Be Aggressive in specifying User Requirements (we are not stating our requirements)
- Always have the user in mind
- Don’t get tied down by technology alone. Technology is changing fast.



Collecting User Requirements

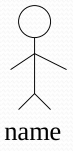
- Identify users
- Identify their roles, responsibilities and needs
- Asking users is not enough observing user in action only can give complete picture of what he needs.
- User - Task Analysis
- Define Problem Statements

Use Case Diagrams

- Use case diagrams are used to visualize, specify, construct, and document the (intended) behavior of the system, during requirements capture and analysis.
- Provide a way for developers, domain experts and end-users to Communicate.
- The same use case diagram can be used in requirement analysis, design and testing.
- Use case diagrams contain use cases, actors, and their relationships.

Actors

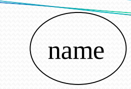
- An actor represents a set of roles that users of the system can play
- Actors can be human or automated systems.
- Actors are entities which require help from the system to perform their task or are needed to execute the system's functions.
  - An actor may actively interchange information with the system
  - An actor may be a passive recipient of information
- Actors are not part of the system.



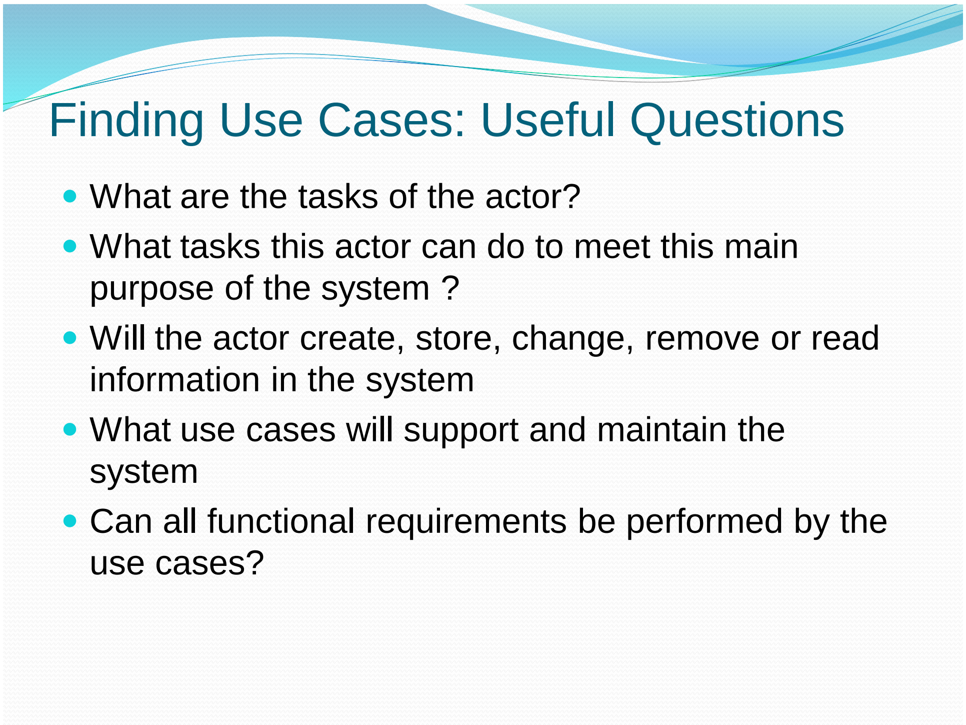
Finding Actors: useful questions

- Who is interested in a certain requirement?
- Who will supply the system with the information, use this information, remove this information?
- Who will use the results of this requirement?
- Does the system use an external resource?
- Does one actor play several different roles? Do several actors play the same role?

Use Case

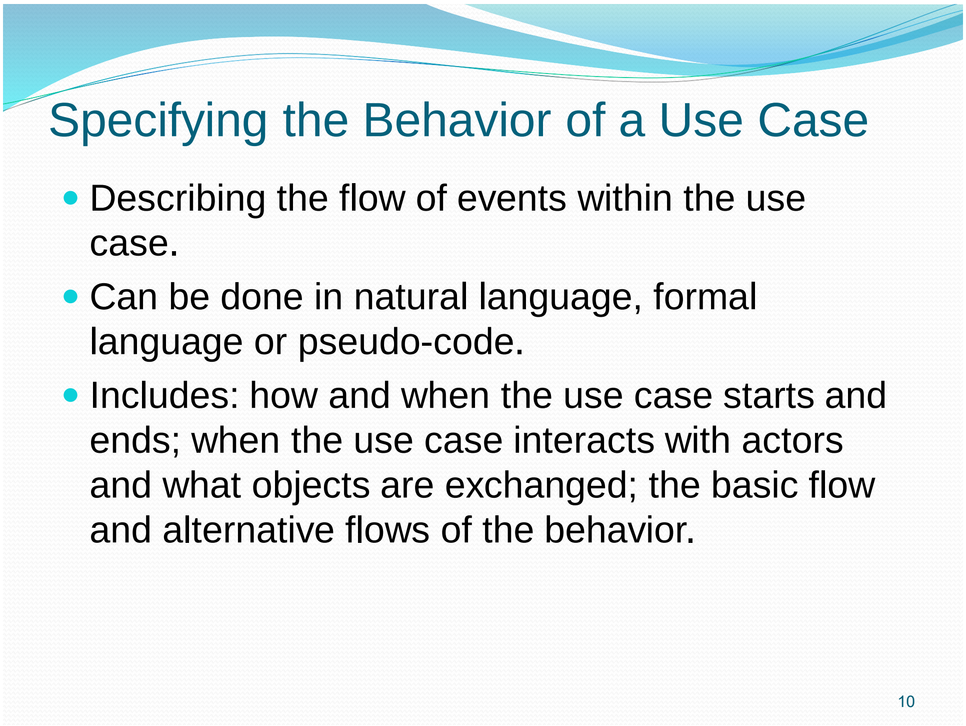


- A use-case is a description of how one of the actors uses the system to accomplish a certain goal
- Use-cases describe in natural language the complete functionality of a system. Each use-case is a snapshot of a particular aspect of a system.
- A use case diagram shows the relationship among actors and use cases within a system.



Finding Use Cases: Useful Questions

- What are the tasks of the actor?
- What tasks this actor can do to meet this main purpose of the system ?
- Will the actor create, store, change, remove or read information in the system
- What use cases will support and maintain the system
- Can all functional requirements be performed by the use cases?



Specifying the Behavior of a Use Case

- Describing the flow of events within the use case.
- Can be done in natural language, formal language or pseudo-code.
- Includes: how and when the use case starts and ends; when the use case interacts with actors and what objects are exchanged; the basic flow and alternative flows of the behavior.

Use Case Description

Each use case may include all or part of the following:

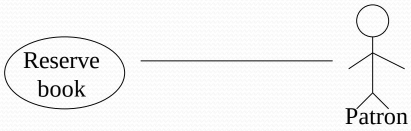
Title or Reference Name	- meaningful name of the UC
Author/Date	- the author and creation date
Modification/Date	- last modification and its date
Purpose	- specifies the goal to be achieved
Overview	- short description of the processes
Cross References	- requirements references
Actors	- agents participating
Pre Conditions	- must be true to allow execution
Post Conditions normally	- will be set when completes
Normal flow of events	- regular flow of activities
Alternative flow of events	- other flow of activities
Exceptional flow of events	- unusual situations
Implementation issues	- foreseen implementation problems

Use Cases and Actors

- From the perspective of a given actor, a use case does something that is of value to the actor, such as calculate a result or change the state of an object.
- The Actors define the environments in which the system lives

Relationships between Use Cases and Actors

- Actors may be connected to use cases by associations, indicating that the actor and the use case communicate with one another using messages.

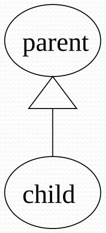


Relationships between Use Cases

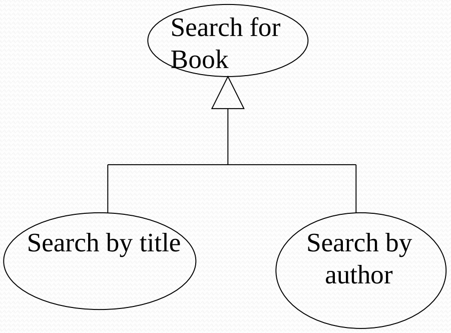
1. Generalization - use cases that are specialized versions of other use cases.
2. Include - use cases that are included as parts of other use cases. Enable to factor common behavior.
3. Extend - use cases that extend the behavior of other core use cases. Enable to factor variants.

Generalization

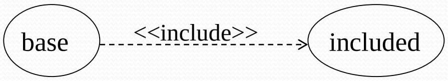
- The child use case inherits the behavior and meaning of the parent use case.
- The child may add to or override the behavior of its parent.



More about Generalization

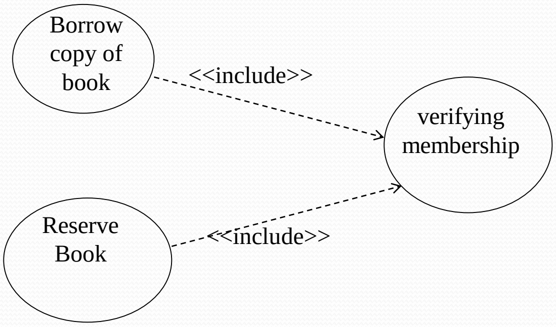


Include

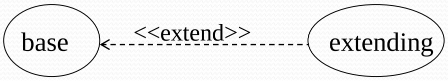


- The base use case explicitly incorporates the behavior of another use case at a location specified in the base.
- The included use case never stands alone. It only occurs as a part of some larger base that includes it.

More about Include

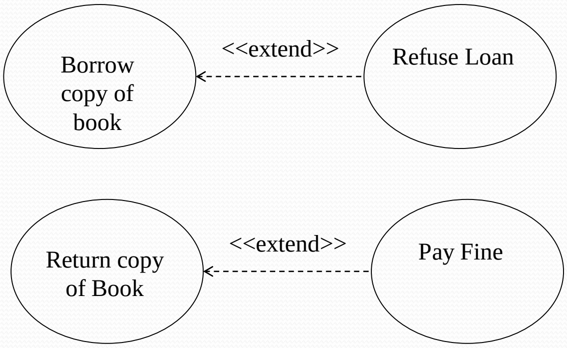


Extend



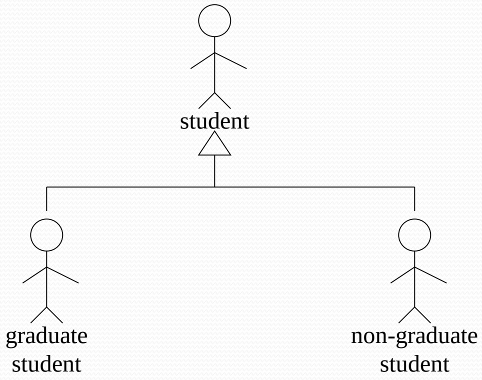
- The base use case implicitly incorporates the behavior of another use case at certain points called extension points.
- The base use case may stand alone, but under certain conditions its behavior may be extended by the behavior of another use case.

More about Extend



Relationships between Actors

- Generalization.



student

graduate student

non-graduate student

Example - Library System

- CF(Customer Satisfaction)=data+system+services
- CF for the Data
  - Comprehensiveness of information
  - Freshness of Information
  - Reliability of Information
- CF for the System
  - User Friendliness of Interface
  - Efficiency of retrievals
  - Accessibility
- CF for the additional Services
  - Copy and Social activities

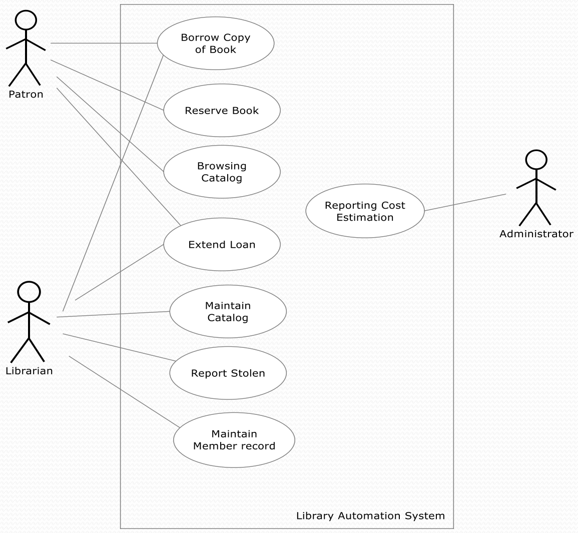
Objectives of Library System

- Reduction of Management Cost
  - Extension of automatic services
  - Automatic data collection and reporting
- Collaboration of other libraries
  - Sharing database
  - Union catalog
  - Partial Responsibility of collections

Library System

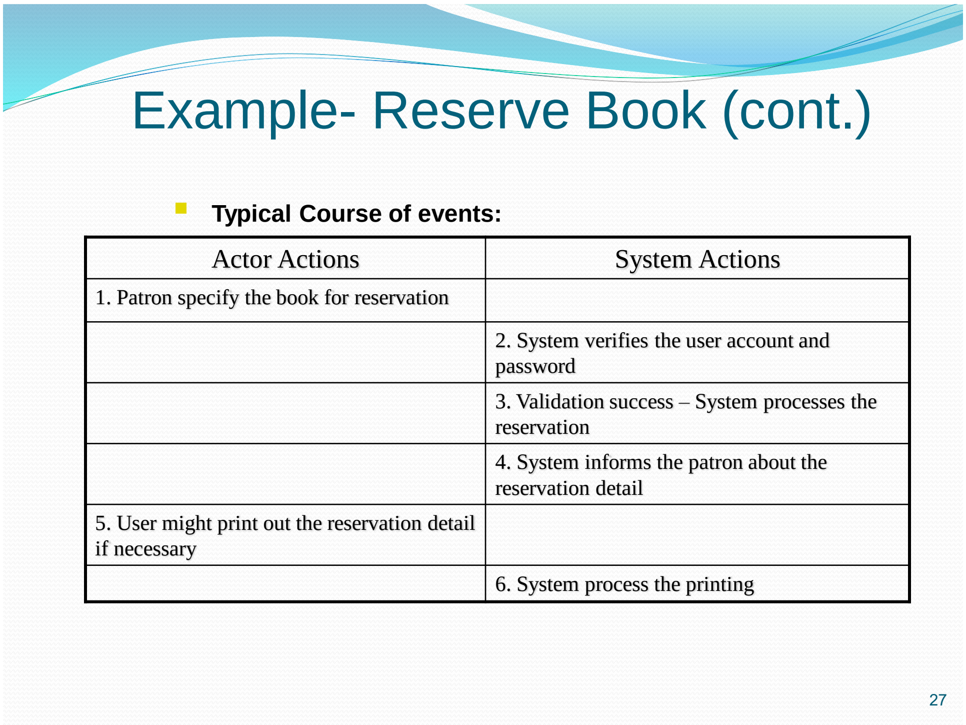
- Users (Actors)
  - Patron (Teacher, Student and others)
  - Librarian
  - Book Seller
  - Publisher
  - Administrator

Library System



Example- Reserve Book

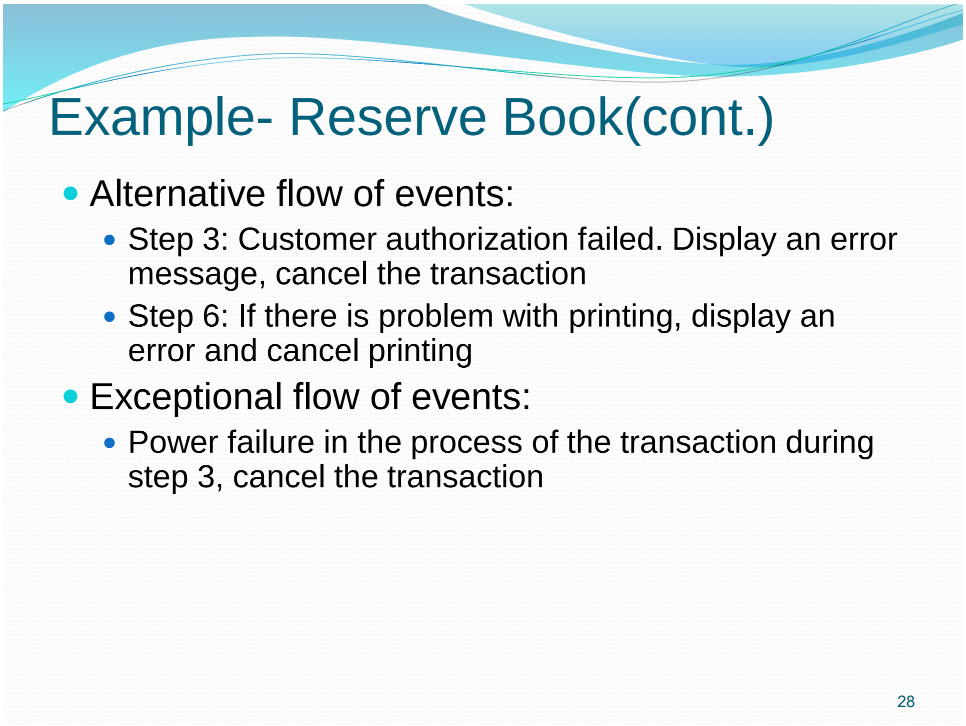
- Use Case: Reserve Book
- Author : Patron
- Date: 28-Aug -2009
- Purpose: To reserve a book which is currently not available for loan
- Overview: The use case start when the patron specify the book for reservation. The system request the account and password. The system validates the account information. If the validation succeeded, the system accept the reservation and display the reservation detail (Current position in the waiting list of reservation, Expect date of availability) for the patron. Otherwise – validation failure is executed.
- Actor: Patron
- Pre-condition:
  - The book is currently not available for loan
- Post-condition:
  - The reservation list of the book is updated
  - The reservation detail is recorded



Example- Reserve Book (cont.)

- **Typical Course of events:**

Actor Actions	System Actions
1. Patron specify the book for reservation	
	2. System verifies the user account and password
	3. Validation success – System processes the reservation
	4. System informs the patron about the reservation detail
5. User might print out the reservation detail if necessary	
	6. System process the printing



Example- Reserve Book(cont.)

- Alternative flow of events:
  - Step 3: Customer authorization failed. Display an error message, cancel the transaction
  - Step 6: If there is problem with printing, display an error and cancel printing
- Exceptional flow of events:
  - Power failure in the process of the transaction during step 3, cancel the transaction

More Example – Bookshop Management