Object Oriented System Analysis and Design

Chapter Three: Gathering User Requirements

Ewnetu E. (MSC.)

May 27, 2023

Contents

- Requirement gathering
- Requirement gathering Process
- Requirement gathering techniques
- Usecase modeling
- UI Prototyping

What is Requirement Elicitation ?

- Requirements elicitation: is the practice of collecting different requirements of the system from different requirement sources.
- It is just the process of extracting the information from users, customers, or group of people, existing systems or written documents.
- It is all about learning and understanding the needs of users and project sponsors with the ultimate aim of communicating their needs to the analysts and developers.
- A substantial part of elicitation is dedicated to uncovering, extracting, and surfacing the needs/wants of the potential stakeholders.

...Cont'd

- Requirement Gathering:
  - Is a process of identifying needs of the system (input specification).
  - Helps as the primary input to analysis phase.
  - Requirements are front ends to systems development.
  - Helps to identify the relevant parties, usually the stakeholders.
  - Helps to gather the WishList from each stakeholder.
  - Helps to document and refine the WishList.
- Expected properties of requirements
  - Unambiguous: not open to more than one interpretation.
  - Complete: having all the necessary or appropriate parts.
  - Verifiable: can be tested and proven to be true.
  - Consistent: done in the same way over time, especially so as to be fair or accurate.
  - Modifiable: to change somewhat the form, improve qualities, alter partially, amend.
  - Traceable: can be tracked or detected.

Types of Requirements

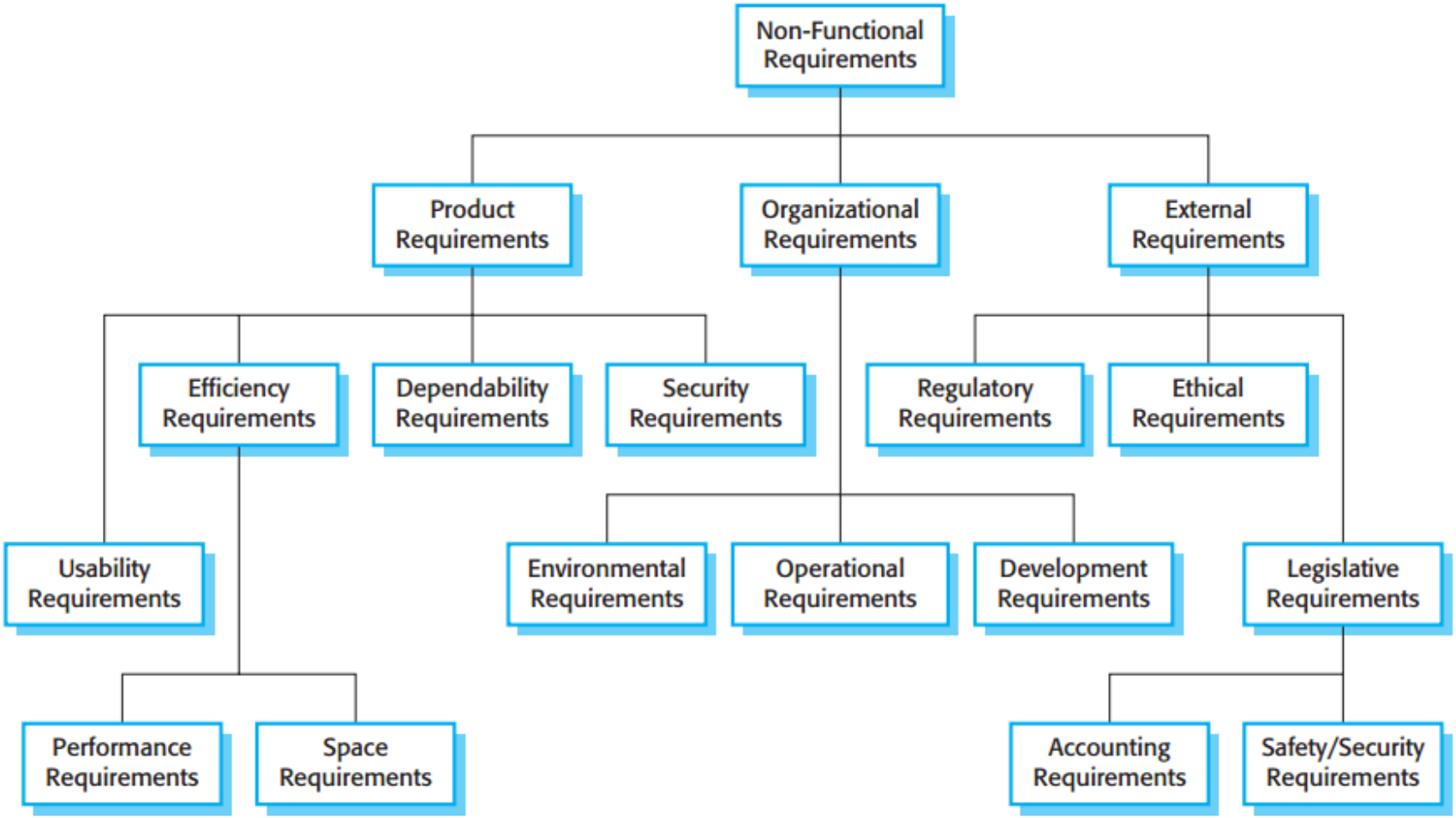
- Functional requirements:
  - An action that a system must be able to do or perform.
  - An important category of the real requirements.
  - Sometimes called behavioral or operational requirements .
  - They specify :
    - ★ The inputs (stimuli) to the system,
    - ★ The outputs (responses) from the system, and behavioral relationships between them.
- Non-functional requirements
  - Usability,
  - Performance,
  - Response time,
  - Scalability,
  - Throughput,
  - Availability,
  - Robustness, UI,
  - Supportability: Adaptability, Maintainability

Classification of non-functional Requirements

- Product requirements:
  - Specify that the delivered product must behave in a particular way.
  - e.g.) The user interface for LIBSYS shall be implemented as simple HTML without frames or Java applets.
- Organizational requirements:
  - Which are a consequence of organizational policies and procedures like process standards used, implementation requirements, etc.
  - e.g.) The system development process and deliverable documents shall conform to the processes and deliverables.
- External requirements:
  - Requirements which arise from factors which are external to the system and its development process.
  - e.g.) interoperability requirements, legislative requirements, etc.

...Cont'd

- Detailed classification



Derived requirements

- A derived requirement is a requirement that is further refined from a higher-level requirement or a requirement that results from choosing a specific implementation or system element.
- Done starting from requirements analysis to design phase.

Requirement Measures

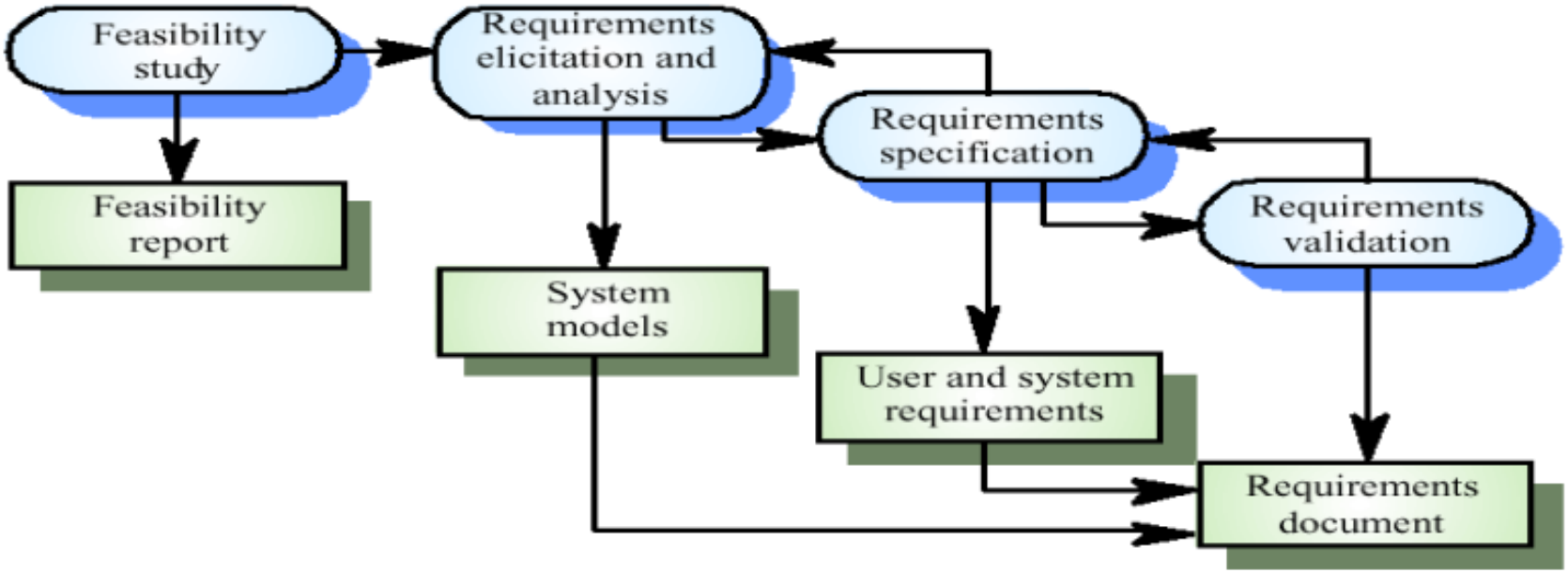
Property	Measure
Speed	Processed transactions/second User/event response time Screen refresh time
Size	Mbytes Number of ROM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems

Requirements Elicitation Process

- The requirements elicitation process involves a set of activities that must allow for communication, prioritization, negotiation, and collaboration with all the relevant stakeholders.
- It is a multi way process
- Many individuals take part in the elicitation process
- Many techniques are conducted
- Many sources are used.
- Requirements elicitation involves activities that are intensely communicative.
- The process of requirements elicitation is generally accepted as one of the critical activities in the RE process.
- Getting the right requirements is considered as a vital but difficult part of software development projects.
- Requirements elicitation can be broken down into the activities of information gathering, fact-finding and integration.

The Requirements Engineering Process

- Processes used to discover, analyze and validate system requirements
- At the end of the requirement engineering process we produce a fully organized requirement document.



Actors in Requirements Engineering Process

- Domain expert: provide information about the application domain and the specific problems in that domain which are to be solved.
- System end-user: will use the system after delivery.
- Requirements engineer: eliciting and specifying the requirements is accomplished through these actors.
- Software engineer: responsible for developing the software /system and make system model.
- Project manager: plan and estimate the whole project from start to end.

Feasibility studies

- A feasibility study decides whether the proposed system is worthwhile or not.
- It is an assessment of the practicality of a proposed project.
- It is a short focused study that checks;
  - If the system contributes to organizational objectives.
  - If the system can be engineered using current technology and within budget.
  - If the system can be integrated with other systems that are used.

Feasibility study types

- Economic feasibility: assessment is to determine the positive economic benefits to the organization that the proposed system will provide.
- Legal feasibility: determines whether the proposed system conflicts with legal requirements like any data protection act or any social media law.
- Technical feasibility: checks whether the developers have practical competence to do the project.
- Operational feasibility: is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.
- Schedule Feasibility: focuses on estimating the start and completion of the project, whether it complies with the schedule and if it deviates out of the schedule.

Elicitation and Analysis

- Involves technical staff working with customers:
- Helps to find out about the application domain, the services that the system should provide and the system's operational constraints.
- May involve:
  - End-users,
  - Managers,
  - Engineers involved in maintenance,
  - Domain experts,
  - Developers.....

Requirement Elicitation Activities

- Requirements elicitation itself is a very complex process involving many activities, with multiple techniques available to perform those activities.
- Typical activities of the requirements elicitation process can be divided into six fundamental classes:
  - Application Domain Understanding
  - Problem Understanding
  - Business Understanding
  - Stakeholders Understanding
  - Identifying the sources of Requirements
  - Selecting the appropriate elicitation technique/approach

...Cont'd

- Application domain understanding: application domain knowledge is knowledge of the general area where the system is applied.
- Problem understanding: the details of the specific customer problem where the system will be applied must be understood.
- Business understanding: you must understand how systems interact and contribute to overall business goals.
- Understanding the needs and constraints of system stakeholders: you must understand, in detail, the specific needs of people who require system support in their work.
- Identifying the sources of requirements
- In all software development projects a number of possible sources for requirements may be identified.
  - Stakeholders
  - Written Documents
  - Existing Systems

...Cont'd

- Stakeholders: represent the most obvious source of requirements.
- Are people who have an interest in the system or are affected in some way by the development and implementation of the system and hence must be consulted during requirements elicitation.
- Typically stakeholders include groups and individuals internal and external to the organization.
- They may be subject matter experts.
- Users and subject matter experts are used to supply detailed information about the problems and user needs.
- Existing systems and processes:
- Represent another source for eliciting requirements, particularly when the project involves replacing a current or legacy system.
- Written documents:
- Describe about systems and business processes including manuals, forms, and reports $\rightarrow$ can provide useful information.

...Cont'd

- Selecting the techniques, approaches, and tools to use:
- Although some may advocate that just one elicitation technique or a single methodology is sufficient and may be applied to all cases, it is generally accepted that an individual requirements elicitation technique or approach cannot possibly be suitable for all projects.
- The choice of techniques to be employed is dependent on the specific context of the project and is often a critical factor in the success of the elicitation process.

...Cont'd

- Techniques & Approaches for Requirements Elicitation:
  - Interviews
  - Questionnaires
  - Group Work
  - Brainstorming
  - Joint Application Design (JAD)
  - Observation
  - Prototyping
- Note: Read in detail about each requirement elicitation technique !

...Cont'd

- Requirements Document:
  - The document used to communicate the requirements to customers, system, and software engineers is referred to as a requirements document (RD) or specification.
  - This refers to a comprehensive collection of the characteristics of a system and the capabilities it will make available to the users.
  - It provides a detailed analysis of the data the system will be expected to manipulate.
  - It may also include a detailed definition of the user interfaces of the system.

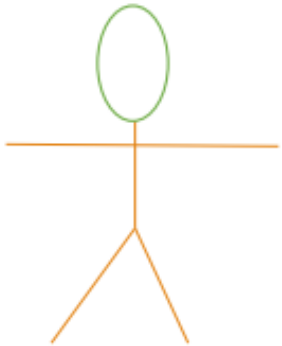
Use case Modeling

- In software and systems engineering, a use case is a list of functions or events, typically defining the interactions between a role (known in UML as an actor) and a system functionality.
- The actor can be a human or other external system.
- A use case is a function performed by a system that yields a measurable result of values for a particular actor.
- Use case model: a diagram that shows the dialogue between an actor and the system.
- A use-case model is a model of the system's intended functions and its surroundings, and serves as a contract between the customer and the developers.
- It represents the functionality provided by the system; $\rightarrow$ what capabilities will be provided to an actor by the system.

Use case diagrams

- The representation of the user's interaction with the system that shows the relationship between the user and different use cases in which the user is involved.
- Describe the functional behavior of the system as seen by the user.
- Use-cases are a scenario based techniques in the UML which identify the actors in an interaction and describe the interaction itself.
- A set of use cases should describe all possible interactions of actors with the system.

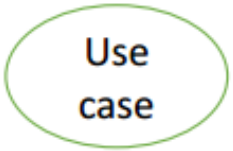
Use case diagrams



Actor: person, external system….

anything that exchanges data with the system, usually external to the system.

Communication link: links the actor and use case.



Use case: A function of value for the actor.

Use case diagrams

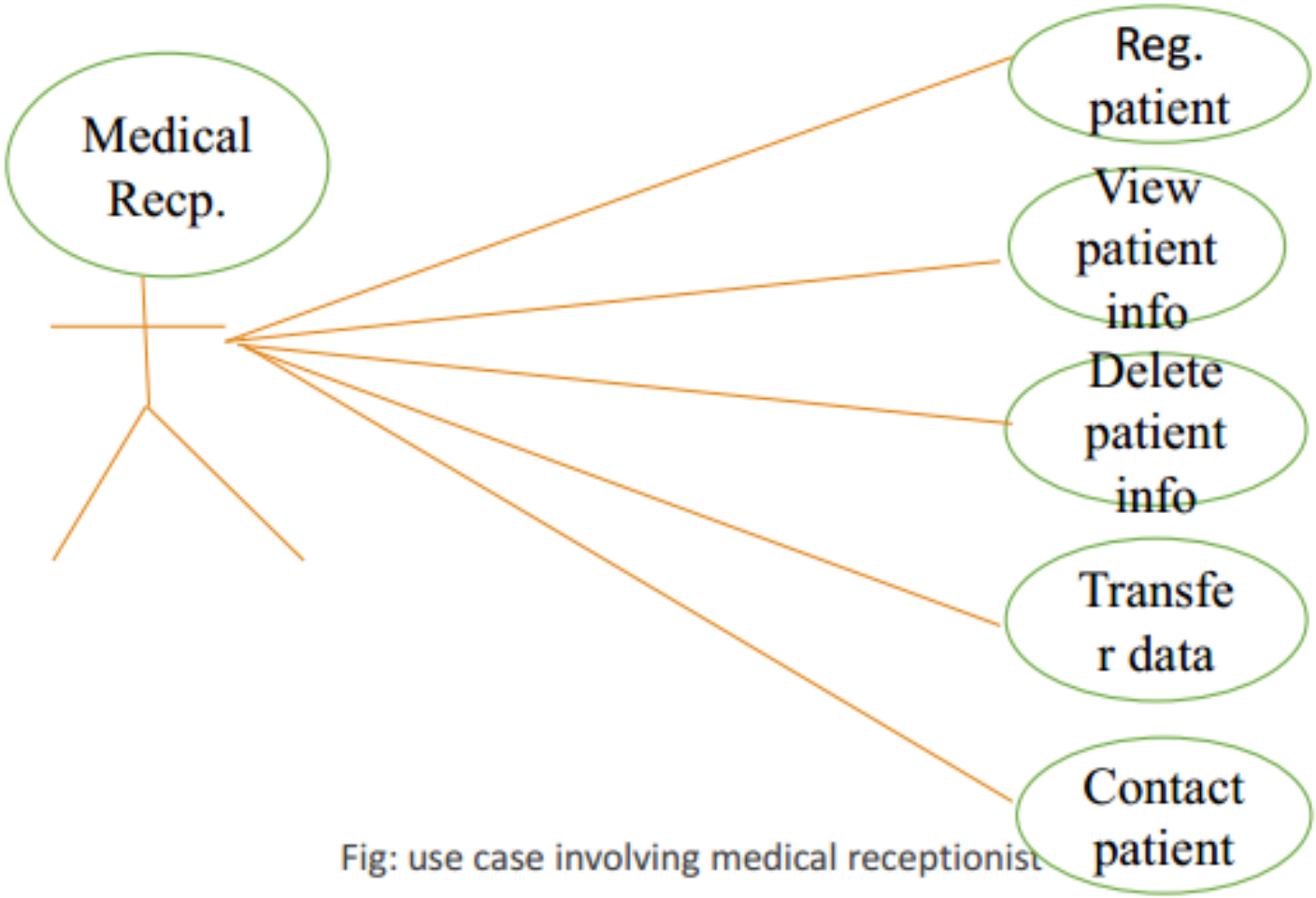
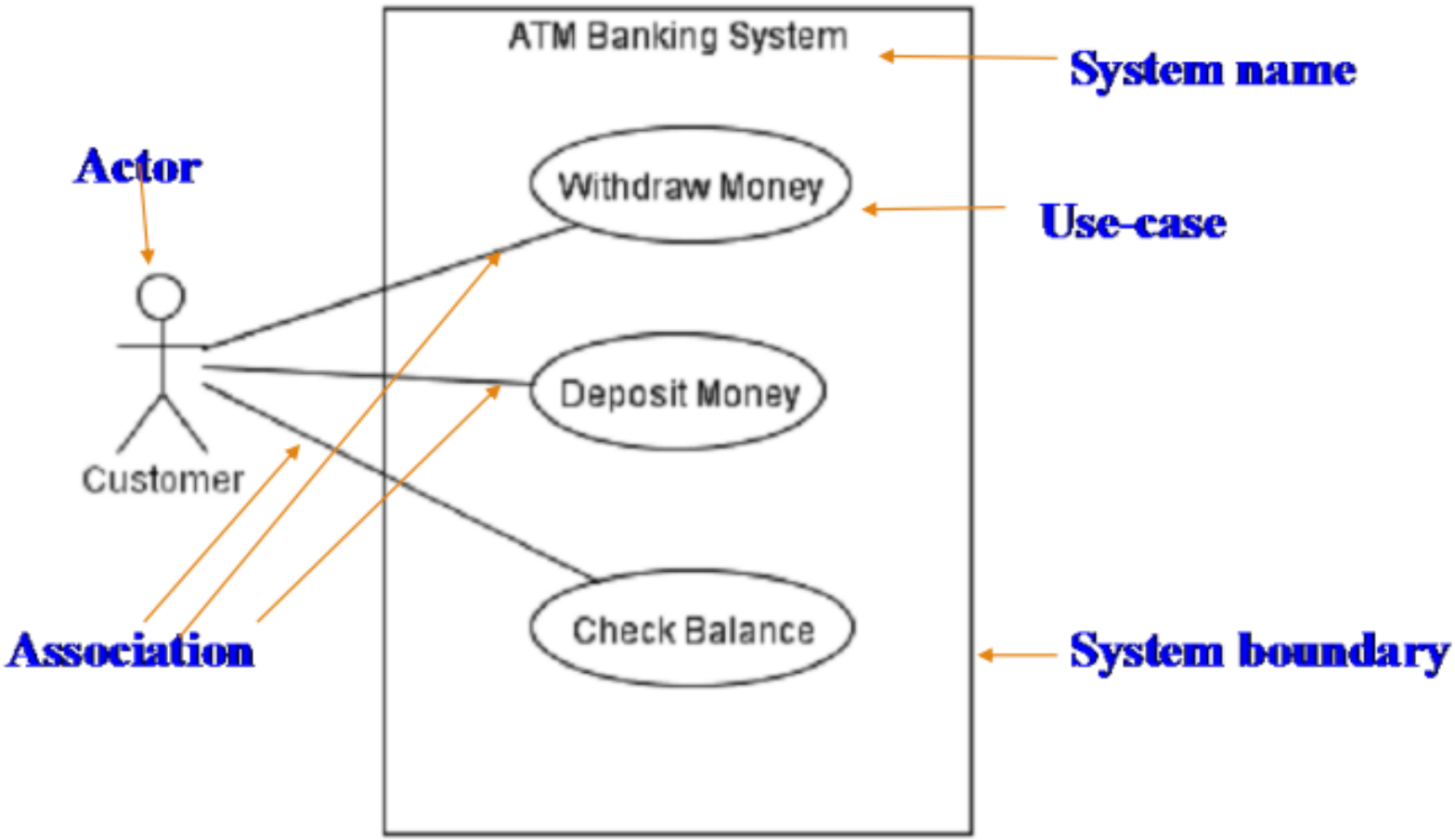
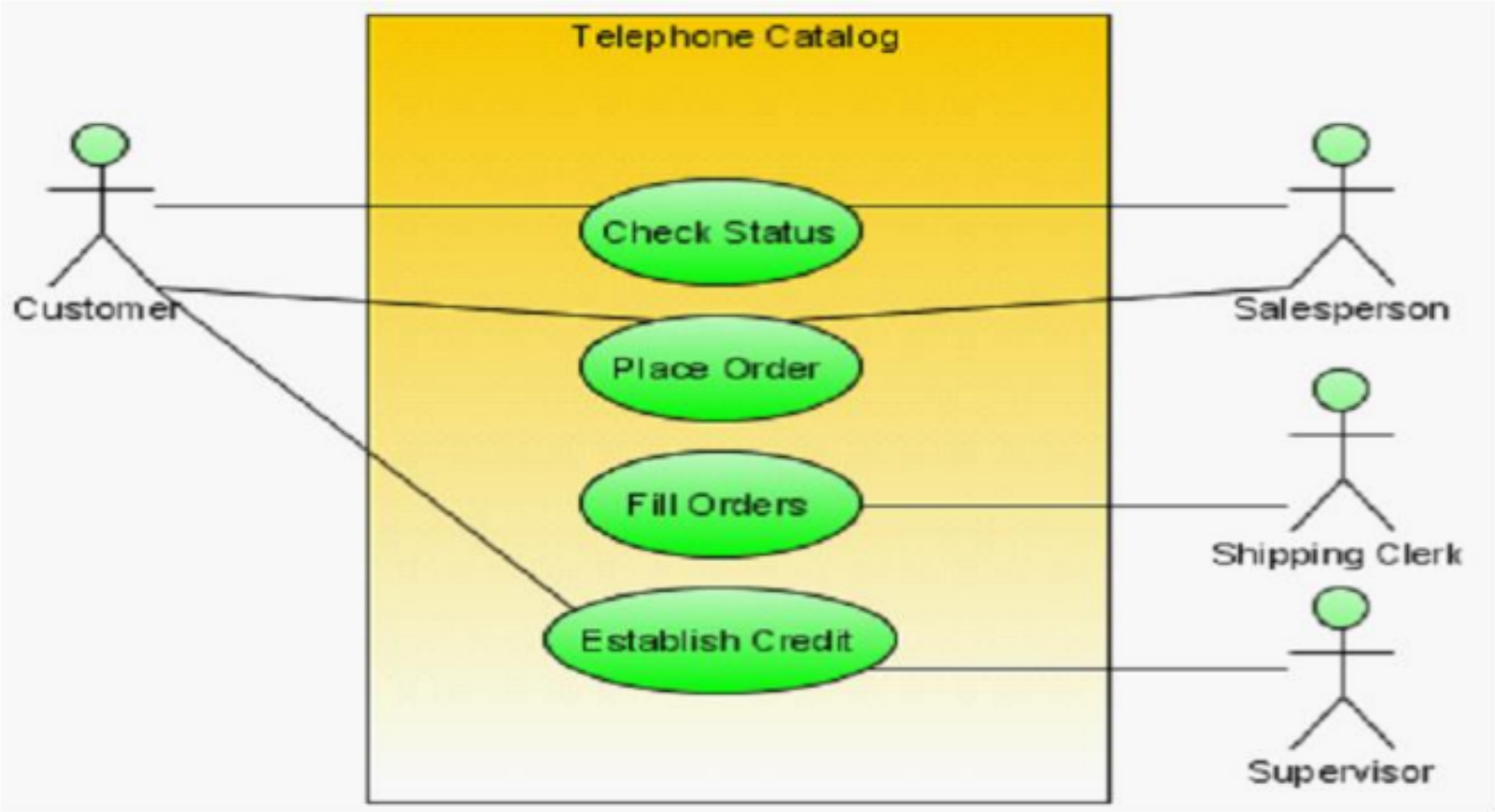


Fig: use case involving medical receptionist

Use case diagrams



Use case diagrams



Scenarios

- Scenarios are descriptions of how a system is used in practice.
- They are helpful in requirements elicitation as people can relate to these more readily than abstract statement of what they require from a system.
- Scenarios are particularly useful for adding detail to an outline requirements description.
- Scenario Descriptions:
  - System state at the beginning of the scenario
  - Normal flow of events in the scenario
  - What can go wrong and how it will be handled
  - Other concurrent activities
  - System state on completion of the scenario

Developing Use case Models of Systems

- Description of types of interactions between a system components (use cases) and external actors.
- Actors: any agent that interact with the system to achieve a useful goal (e.g. people, other software systems, hardware).
- Scenario: describes a typical sequence of actions that an actor performs in order to complete a given task.
- The objective of use case analysis: is to model the system from the point of view of how actors interact with the system when trying to achieve their objectives.

Guidelines for Formulation of Use Cases

- Name
  - Name should indicate what user is trying to accomplish.
  - e.g “RequestMeeting”, “ScheduleMeeting”, “ProposeAlternateDate”
- Use a verb phrase to name use case.
- Length
  - A use case description should not exceed 1-2 pages. If longer, use include relationships.
  - A use case should describe a complete set of interactions.

Guidelines for Formulation of Use Cases

- Flow of events:
  - Use active voice: steps should start either with "The Actor ..." or "The System ".
  - Causal relationship between steps should be clear.
  - All flow of events should be described (not only main flow of event).
  - Boundaries of the system should be clear.
  - Components external to the system should be described as such (scope).
- steps to make use case diagram : Identify scope $\rightarrow$ Identify Actors $\rightarrow$ Identify use cases $\rightarrow$ use links $\rightarrow$ Construct use case diagram

Scenario

- A synthetic description of an event or series of actions and events.
- A textual description of the usage of a system.
- The description is written from an end user's point of view.
- A scenario can include text, video, pictures and story boards.
- A scenario is an instance of a use case.
- It expresses a specific occurrence of the use case (a specific path through the use case): $\rightarrow$ A specific actor, At a specific time, With specific data ...
- Many scenarios may be generated from a single use case diagram.
- Each scenario may require many test cases.

Scenario

- Scenarios are stories which explain how a system might be used.
- It should include;
  - Description of the system state before entering the scenario,
  - The normal flow of events in the scenario,
  - Exceptions to the normal flow of events,
  - Information about concurrent activities,
  - Description of the system state at the end of the scenario.

User interface Prototyping

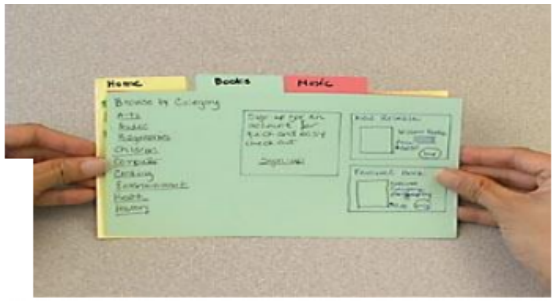
- User interface (UI) is the portion of software with which a user directly interacts.
- An essential user interface prototype also known as an abstract prototype is a low-fidelity model, of the UI for your system.
- At requirement analysis phase, it represents the general ideas behind the UI, but not the exact details.

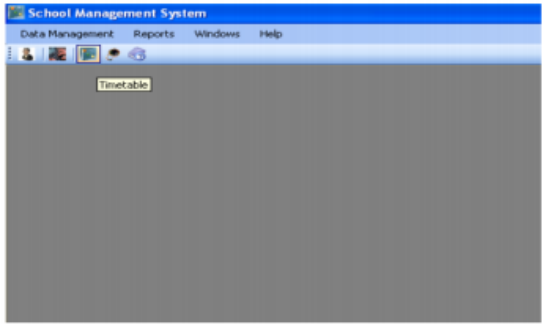
Prototype Example

Prototypes can take many forms:

- ✓Paper prototypes (see http://www.paperprototyping.com/)
- ✓Prototype on index card
- ✓Storyboard/secnarios
- ✓Screen mock-ups
- ✓Models (executables)

Login Screen





Thank you!!!