
بسم الله الرحمن الرحيم

Web Technologies and Programming

Lecture 03

Requirement Engineering

Modeling web applications

Summary of the previous lecture

- **Development Process model**
  - **software development process activities**
- **Requirement for a web development process model**
- **Rational unified process model (RUP)**
  - **suitability for web application development**

Summary of the previous lecture

- **Project management**
- **Responsibilities/tasks of a Project manager**
  - **Planning**
  - **Risk management**
  - **People management**
  - **Reporting**
  - **Proposal writing**
- **Traditional vs. web project management**

Outline

- Introduction to RE
- RE basics
- Requirements specification
- RE process
- RE specifics in web engineering

- System modeling
- Modeling requirements

1. Requirement Engineering

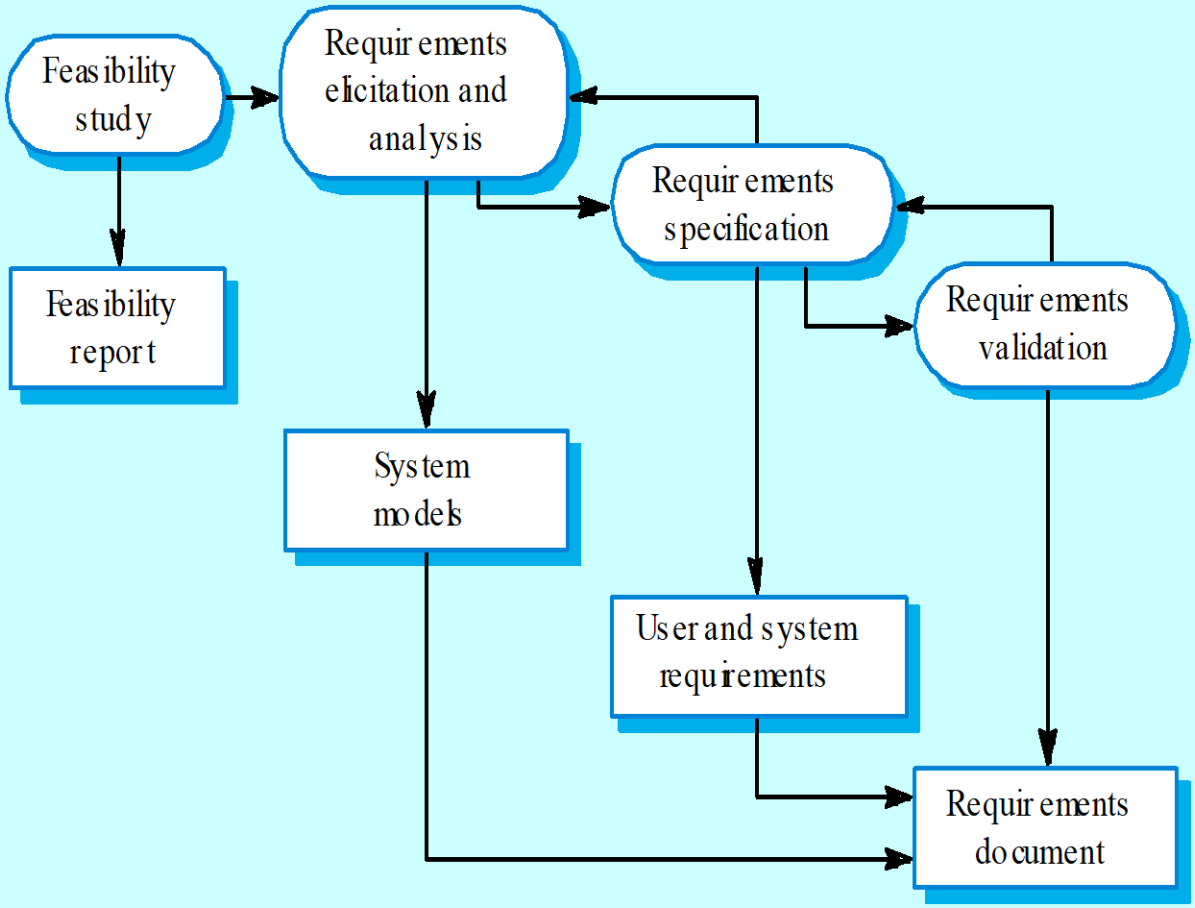
- **Requirements Engineering: the principles, methods, & tools for drawing, describing, validating, and managing project goals and needs**
- **Given the complexity of Web apps, RE is a critical initial stage activity, but often poorly executed**

1. Requirement Engineering

- **It may range from a high-level abstract statement of a service or of a system constraint to a detailed mathematical functional specification.**
- **The processes used for RE vary widely depending on the application domain, the people involved and the organisation developing the requirements.**
- **However, there are a number of generic activities common to all processes**
  - **Requirements elicitation;**
  - **Requirements analysis;**
  - **Requirements validation;**
  - **Requirements management**

1. Requirement Engineering

The requirements engineering process



1. Requirement Engineering…

- **Costs:**
  - **Inadequate software architectures**
  - **“Unforeseen” problems**
    - **budget overruns**
    - **production delays**
    - **“that’s not what I asked for”**
  - **Low user acceptance**

1. Requirement Engineering…

- **Why requirement engineering:**
  - **requirements don't define themselves (Bell & Thayer, 1976)**
  - **removal of mistakes post hoc is up to 200 times more costly (Boehm, 1981)**
  - **iterative collection and refinement is the most important function of a software engineer (Brooks, 1987)**

1. Requirement Engineering…

- **Why requirement engineering:**
  - **A study based on 340 companies in Austria, more than two thirds consider the SRS as the major problem in development process (1995)**
  - **A study on Web applications, 16% systems fully meet their requirement while 53% deployed systems do not (Cutter Consortium, 2000)**

1. Requirement Engineering…

- **Why requirement engineering:**
  - **A study among 8000 projects, 30% of projects fail before completion & almost half do not meet customer requirements (Standish group, 1994)**
    - **Unclear objectives, unrealistic schedules & expectations, poor user participation**

2. RE basics

- **Identify and involve the stakeholders**
  - **those that directly influence the requirements**
  - **customers, users, developers**
- **What are their expectations?**
  - **may be misaligned or in conflict**
  - **may be too narrowly focused or unrealistic**

2. RE basics…

- What is requirement?
- The descriptions of what the system should do
  - services that it provides and the constraints on its operation
- IEEE 601.12 definition of requirement:
  - 1) Solves a user's problem
  - 2) Must be met or possessed by the system to satisfy a formal agreement
  - 3) Documented representation of conditions in 1 and 2

2. RE basics…

- **Requirements types**
- **Functional requirements:**
  - **statement of services**
  - **how system reacts to input**
  - **how system behaves in particular situation**
- **Non-functional requirements:**
  - **constraints on services (timing, quality etc.)**
  - **applies as a whole**

2. RE basics…

- **Requirements are collected iteratively and change**
- **Keys to requirement definition:**
  - **Negotiation**
  - **Scenario-based discovery**
  - **Clear definition of context and constraints**

3. Requirements specifications

- process of writing down the user and system requirements in a requirements document
- User requirements (for users)
  - who do not have a technical background.
  - should be understand able to users
  - avoid notations, use simple tables, forms etc.
- System requirements (for Software engineers)
  - starting point for the system design
  - how system provides the services
  - include more technical information.

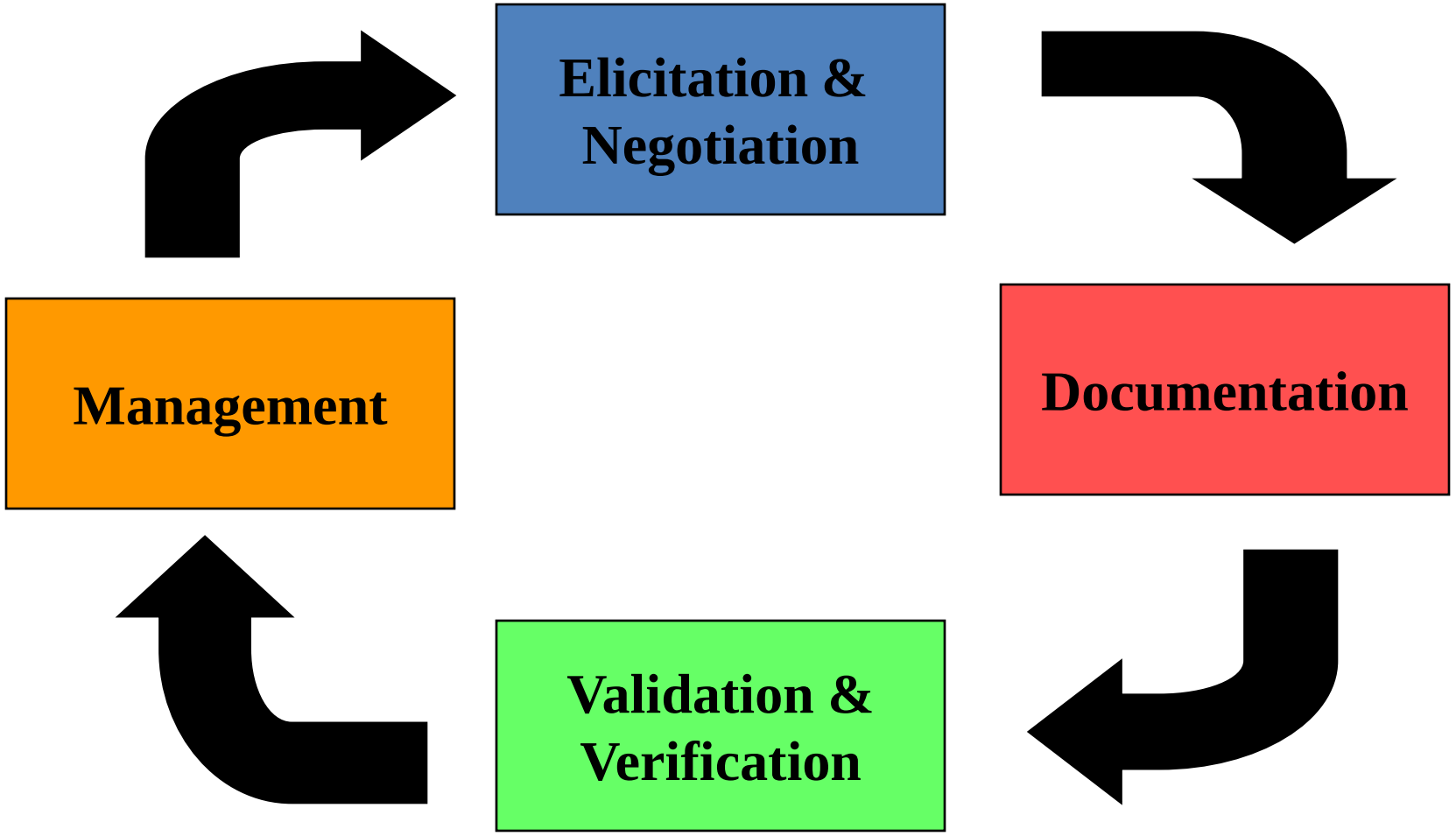
3. Requirements specifications…

- **Natural language specification:**
- **Stories or itemized requirements**
  - **create a standard format**
  - **distinguish between mandatory and desirable requirements**
  - **don’t use the technical words**
  - **associate rationale with each requirement**

3. Requirements specifications…

- **Structured specification:**
- **Includes**
  - **description**
  - **inputs/outputs**
  - **description of the action**
  - **pre condition**
  - **post condition**

4. RE process


Elicitation & Negotiation
Documentation
Validation & Verification
Management

4. RE process…

- **Elicitation and negotiation:**
- **RE engineer involve the stakeholder to define**
  - **application domain**
  - **services**
  - **constraints**
- **Steps:**
  - **requirement discovery**
    - **Interviewing, scenarios, questionnaires, use-cases etc.**
  - **classification and organization**
  - **prioritization and negotiation**

4. RE process…

- **Documentation:**
  - **requirements are documented after consensus**
- **Requirement verification and validation:**
  - **validated: doing right things?**
  - **verification: doing things right?**

4. RE process…

- **Requirements management:**
- Requirements management is the process of managing changing requirements during the requirements engineering process and system development.

- **understanding and controlling changes**
  - **problem analysis and change specification**
  - **change analysis and costing**
  - **change implementation**

5. RE specifics in web

- **Distinguishing characteristics:**
- **Multidisciplinary:**
  - **experts from different disciplines i.e. media experts, content experts, usability experts etc.**
  - **challenging to achieve consensus**
- **Unavailability of stakeholders:**
  - **many stakeholders such as users are unknown during RE process**
  - **need to find suitable representatives**

5. RE specifics in web

- **Distinguishing characteristics:**
- **Rapidly changing requirements & constraints:**
  - **environment is highly dynamic**
  - **harder to stabilize requirements**
- **Unpredictable operational environment:**
  - **impossible to control the operation environment**
  - **affects the quality requirements**
    - **change of bandwidth can change response time**

5. RE specifics in web

- **Distinguishing characteristics:**
- **Legacy Systems:**
  - **constrained by existing system**
  - **existing components drive the possibilities**
- **Quality aspects:**
  - **are decisive i.e. performance, security, availability**
  - **harder to get exact specification**

5. RE specifics in web

- **Distinguishing characteristics:**
- **User interface:**
  - **key success-critical aspect**
  - **should be aware of usability principles**
- **Quality of content:**
  - **accuracy, objectivity, credibility, relevance, actuality, completeness, or clarity**

5.1 RE principles for web

- **Understanding the system context**
  - **web apps are always a component of a larger entity**
  - **why do we need the system?**
  - **how will people use it?**
- **Involving the stakeholders**
  - **get all groups involved**
  - **balance – one group's gain should not come at the expense of another**
  - **repeat the process of identifying, understanding and negotiating**

5.1 RE principles for web

- **Iteratively define requirements**
  - **requirements need to be consistent with other system aspects (UI, content, test cases)**
  - **start with key requirements at a high level; basis for:**
    - **feasible architectures**
    - **key system use cases**
    - **initial plans for the project**

5.1 RE principles for web

- **Risk Orientation**
  - **risk management is at the heart of the analysis process**
  - **what are the greatest risks?**
    - **integration issues / legacy systems**
    - **expected vs. actual system quality**
  - **how to mitigate risks?**
    - **prototyping**
    - **show changes to customer iteratively**
    - **integrate existing systems sooner than later**

Modeling web applications

1. System modeling

- **Process of developing abstract models of a system**
- **Representing system using graphical notations**
  - **UML**

1. System modeling

- each model presents a different view or perspective of the system
  - External perspective: system context and environment
  - Interaction perspective: how system interact with environment or within the system components
  - Structural perspective: how system is organized
  - Behavioral perspective: dynamic behavior of the system and how it responds to events.

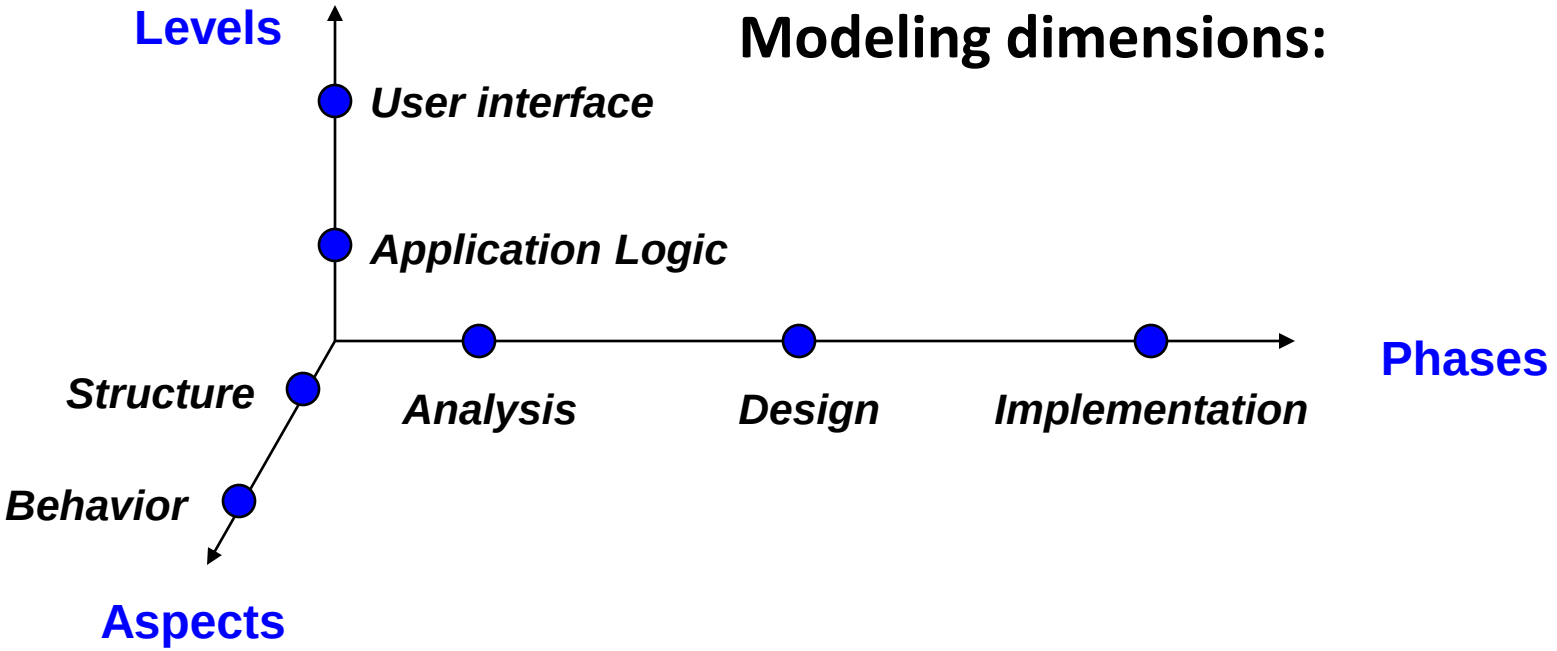
1. System modeling…

- **Models are used during**
  - **RE phase to derive system requirements**
    - **use-case diagram, activity diagram**
  - **design phase to describe the system to engineers**
    - **class diagrams, sequence diagrams etc.**
  - **after implementation**
    - **to document system's structure and operation**

1. System modeling…

- **Why system modeling?**
  - **reduce complexity**
  - **document design decisions**
  - **facilitate communication among team members**

1. System modeling…

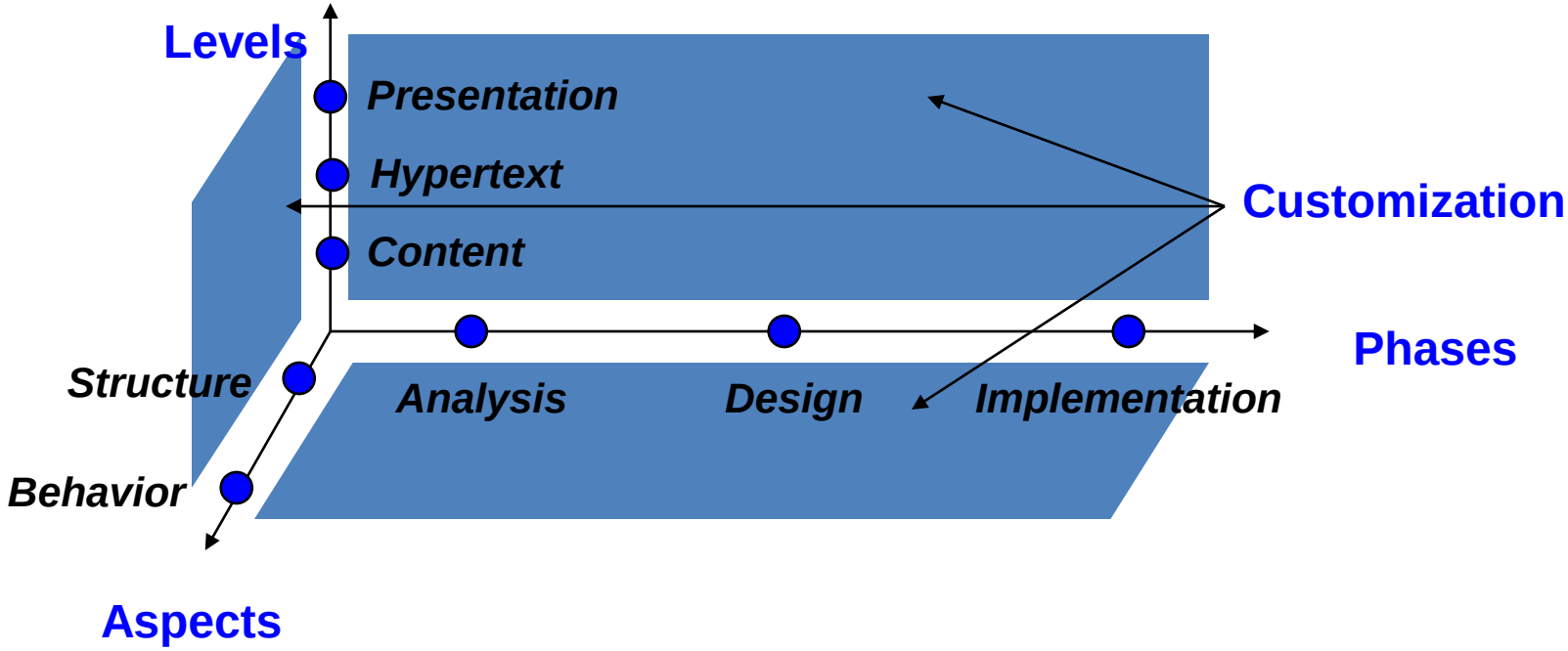


- Levels – the "how" & "what" of an application
- Aspects – objects, attributes, and relationships; function & processes
- Phases – Development cycle

1. System modeling…

- **"The Unified Modeling Language is a visual language for specifying and documenting the artifacts of systems"**
  - **Structural – Class diagrams**
  - **Behavioral – Use Case diagrams, State machine diagrams**

1. System modeling…



- **Levels – Information, node/link structure, UI & page layout separate.**
- **Aspects – Same as Software Applications**
- **Phases – Approach depends upon type of application**
- **Customization – Context information (user's preferences, bandwidth restriction, device characteristic etc.) and allow to adopt web application accordingly**
- **Influence other three dimensions**

1. System modeling…

- **Requirement modeling**
  - use-case diagram
  - activity diagram
- **Content modeling**
  - class diagram
- **Navigational modeling**
  - to model nodes and navigational structure among them
- **Presentation modeling**
  - model user interface, page-layout

1. System modeling…

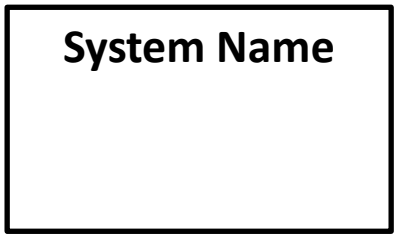
- **For Web-centric modeling, UML is used with some extensions from UWE (UML-based web engineering)**
- **http://uwe.pst.ifi.lmu.de/**

2. Modeling requirements

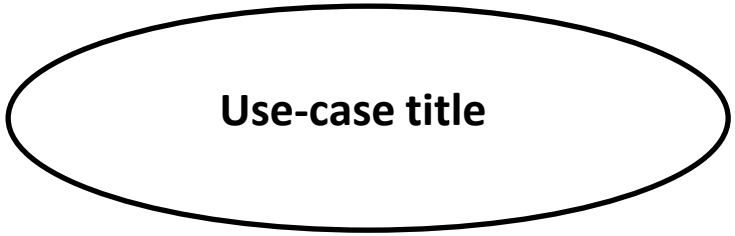
- **Use-case Diagram:** **The goal of the diagram is to provide a high-level explanation of the relationship between the system and the outside world (set goals)**
- **Activity diagram:** **a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency**

2.1 Use-case diagram

- **Components:**
- **The system**

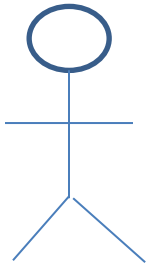


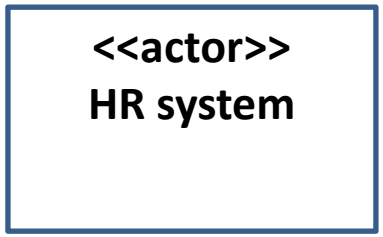
- **The use case task referred to as the use case that represents a feature needed in a software system**



2.1 Use-case diagram

- **Components:**
- **The actor(s) who trigger the use case to activate**

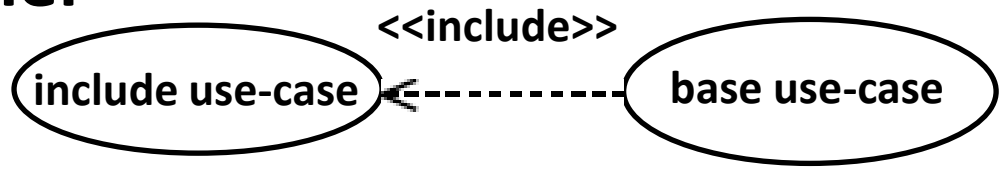




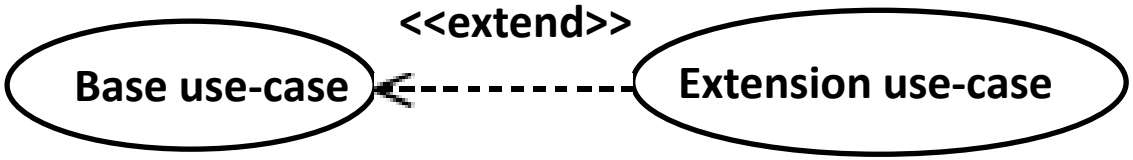
- **The communication line to show how the actors communicate with the use case**

2.1 Use-case diagram…

- **The include relationship represents the inclusion of the functionality of one use case within another**

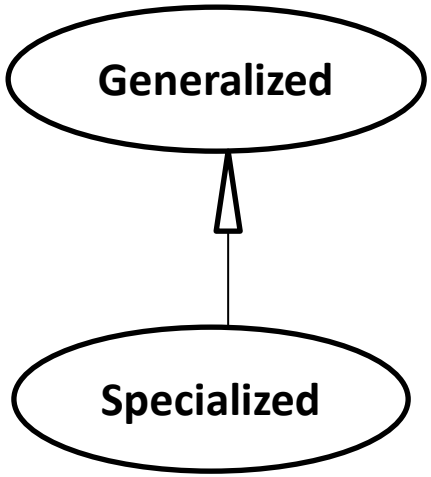


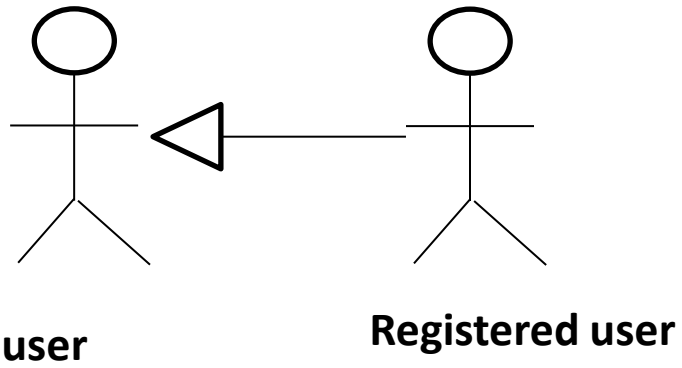
- **The extend relationship represents the extension of the use case to include optional functionality**



2.1 Use-case diagram…

- **A use-case-generalization is a relationship from a child use case to a parent use case, specifying how a child can specialize all behavior and characteristics described for the parent**





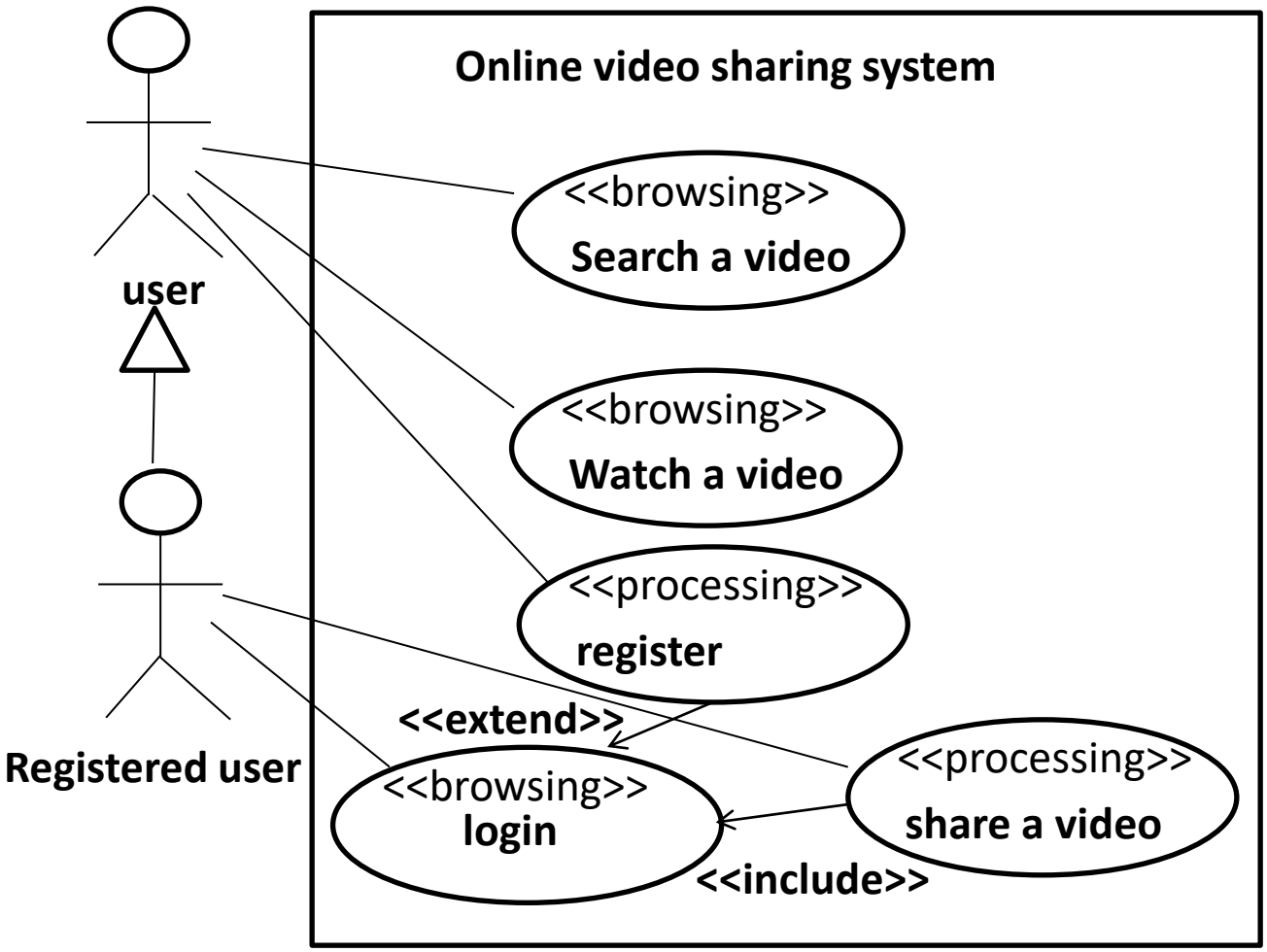
2.1 Use-case diagram…

- **Web specific requirements:**
- **Need to distinguish between functional and navigational use-cases**
  - **UWE provides <<browsing>> to represent a navigational use-case while <<processing>> to represent a functional use-case**

2.1 Use-case diagram…

- **Consider an online video sharing system:**
  - **Users can search and view the videos**
  - **A user must be a register user to share videos**

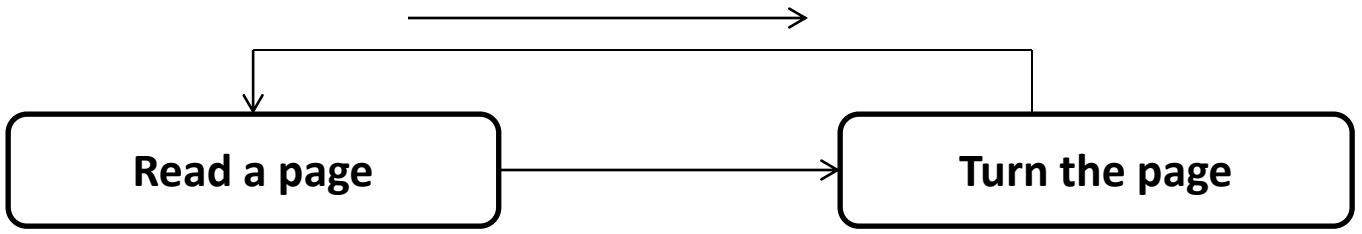
2.1 Use-case diagram…



2. The activity diagram

- **Elements of an activity diagram:**
- **An activity is a step in a process where some work is getting done**

- **The transition takes place because the activity is completed**



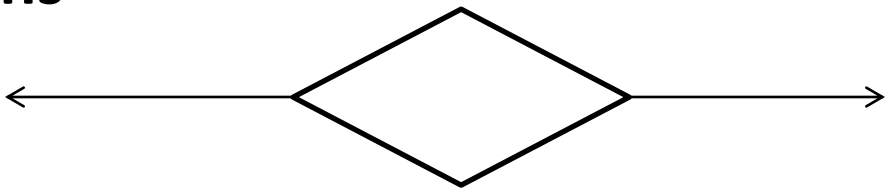
2. The activity diagram

- **Elements of an activity diagram:**

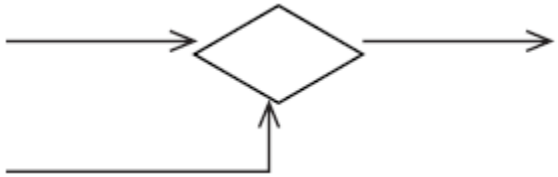
- **A guard condition can be assigned to a transition to restrict use of the transition**

2. The activity diagram…

- **Decisions**

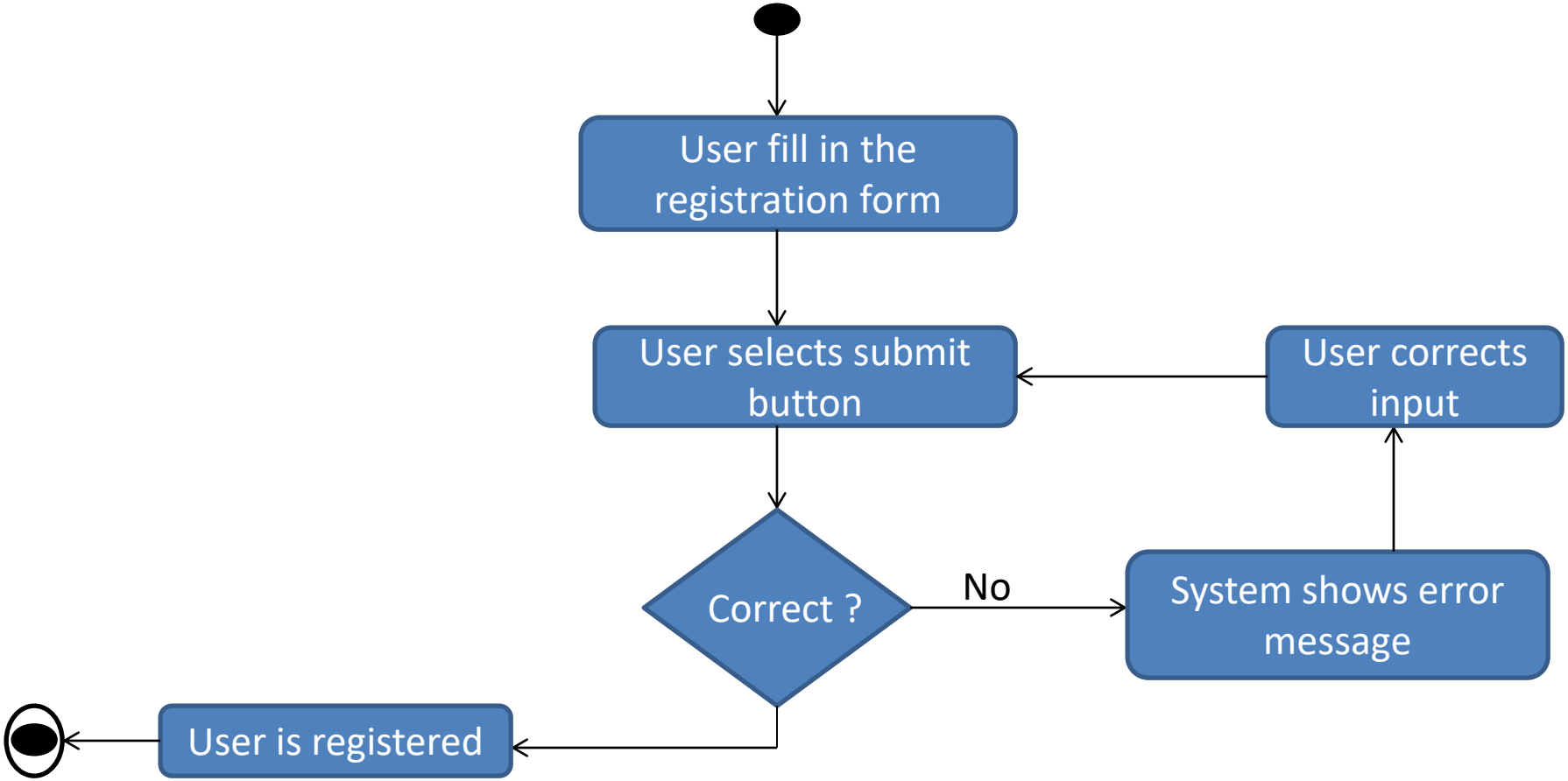


- **Merge point**

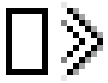


- **Start and end**

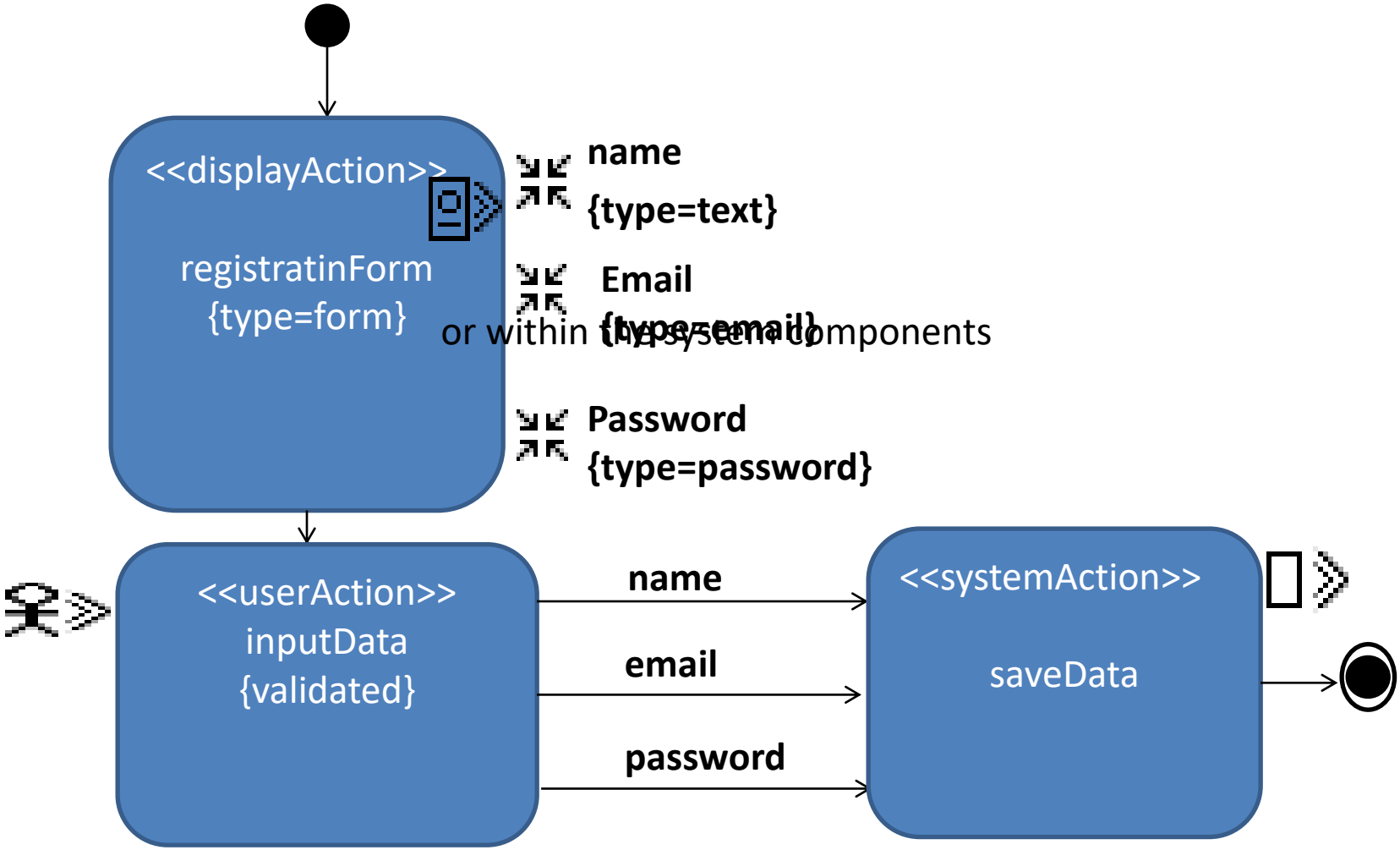
2. The activity diagram…



2. The activity diagram…

- **UWE activity diagram elements:**
- **userAction**  : user's action or response
- **systemAction**  : system's action
- **displayAction**  : display action
- **navigationAction**  : navigation
- **displayPin**  : output
- **interactionPin**  : input

2. The activity diagram…



SUMMARY

- **Introduction to RE**
- **RE basics**
- **Requirements specification**
- **RE process**
- **RE specifics in web engineering**

Summary

- **System modeling**
- **Modeling Requirement**
  - **use-case diagram**
  - **activity diagram**

THANK YOU