

CSCI 4050/6050
Software Engineering

Requirements Elicitation and Specification

Scenarios

1

- **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. It usually also contains details about the work place, social situations and resource constraints.

More Definitions

- **Scenario**: “A narrative description of what people do and experience as they try to make use of computer systems and applications” [M. Carroll, Scenario-Based Design, Wiley, 1995]
- A concrete, focused, informal description of a single feature of the system used by a single actor.

Scenario Example

Scenario: Jim withdraws \$100 from his checking account

1. Jim presses the "Withdrawal" button
2. The system displays the screen to enter the amount of the withdrawal
3. Jim presses digit buttons "1", "0", and "0" to indicate the amount of the withdrawal and then presses "Done"
4. The system displays the screen with Jim's checking and savings accounts to select the source for the withdrawal
5. Jim presses the "Checking" account button
6. The system displays the screen to wait for the cash dispenser; then the cash is dispensed in the cash slot
7. Jim collects the cash

Scenario-Based Design

Scenarios can have many different uses during the software lifecycle

- ***Requirements Elicitation***: As-is scenario, visionary scenario
- ***Client Acceptance Test***: Evaluation scenario
- ***System Deployment***: Training scenario

Types of Scenarios

- **As-is scenario:**
 - Describes a current situation. Usually employed in re-engineering projects. The user describes the system
 - **Example:** Description of Mary's transferring money between two bank accounts
- **Visionary scenario:**
 - Describes a future system. Usually used in greenfield engineering and reengineering projects
 - Can often not be done by the user or developer alone
 - **Example:** Description of an interactive Web-based Tic-Tac-Toe game tournament

Additional Types of Scenarios

- **Evaluation scenario:**
 - Description of a user task against which the system is to be evaluated.
 - **Example:** Four users (two novice, two experts) play in a Tic-Tac-Toe Web-based tournament.
- **Training scenario:**
 - A description of the step by step instructions that guide a novice user through a system
 - **Example:** How to play Tic-Tac-Toe in the on-line setting.

How do we find scenarios?

- Don't expect the client to be verbal if the system does not exist
 - Client understands problem domain, not the solution domain.
- Don't wait for information even if the system exists
 - “What is obvious does not need to be said”
- Engage in a dialectic approach
 - You help the client to formulate the requirements
 - The client helps you to understand the requirements
 - The requirements evolve while the scenarios are being developed

Heuristics for finding scenarios

- Ask yourself or the client the following questions:
 - What are the primary tasks that the system needs to perform?
 - What data will the actor create, store, change, remove or add in the system?
 - What external changes does the system need to know about?
 - What changes or events will the actor of the system need to be informed about?
- However, don't rely on **questions** *and* **questionnaires** alone
- Insist on **task observation** if the system already exists (interface engineering or reengineering)
 - Ask to speak to the end user, not just to the client
 - Expect resistance and try to overcome it.

Scenario example: Warehouse on Fire

- Bob, driving down main street in his patrol car, notices smoke coming out of a warehouse. His partner, Alice, reports the emergency from their car.
- Alice enters the address of the building into her tablet, a brief description of its location (i.e., north west corner), and an emergency level.
- She confirms her input and waits for an acknowledgment.
- John, the dispatcher, is alerted to the emergency by a beep of his workstation. He reviews the information submitted by Alice and acknowledges the report. He allocates a fire unit and sends the estimated arrival time (ETA) to Alice.
- Alice received the acknowledgment and the ETA.

Observations about Warehouse on Fire Scenario

- Concrete scenario
 - Describes a single instance of reporting a fire incident.
 - Does not describe all possible situations in which a fire can be reported.
- Participating actors
 - Bob, Alice and John

After the scenarios are formulated

- Find all the use cases in the scenario that specify all instances of how to report a fire
 - Example: “Report Emergency” in the first paragraph of the scenario is a candidate for a use case
- Describe each of these use cases in more detail
 - Participating actors
 - Describe the entry condition
 - Describe the flow of events
 - Describe the exit condition
 - Describe exceptions
 - Describe nonfunctional requirements
- Functional Modeling (next lecture: Use cases)

Requirements Elicitation: Difficulties and Challenges

- Communicate accurately about the domain and the system
 - People with different backgrounds must collaborate to bridge the gap between end users and developers
 - Client (customer) and end users have **application domain knowledge**
 - Developers have **solution domain knowledge**
- Identify an appropriate system (defining the system boundary)
- Provide an unambiguous specification
- Leave out unintended features

Example of an Ambiguous Specification

During a laser experiment, a laser beam was directed from earth to a mirror on the Space Shuttle Discovery.

The laser beam was supposed to be reflected back towards a mountain top 10,023 feet high.

The operator entered the elevation as “10023”.

The light beam never hit the mountain top
What was the problem?

Example of an Ambiguous Specification

During a laser experiment, a laser beam was directed from earth to a mirror on the Space Shuttle Discovery.

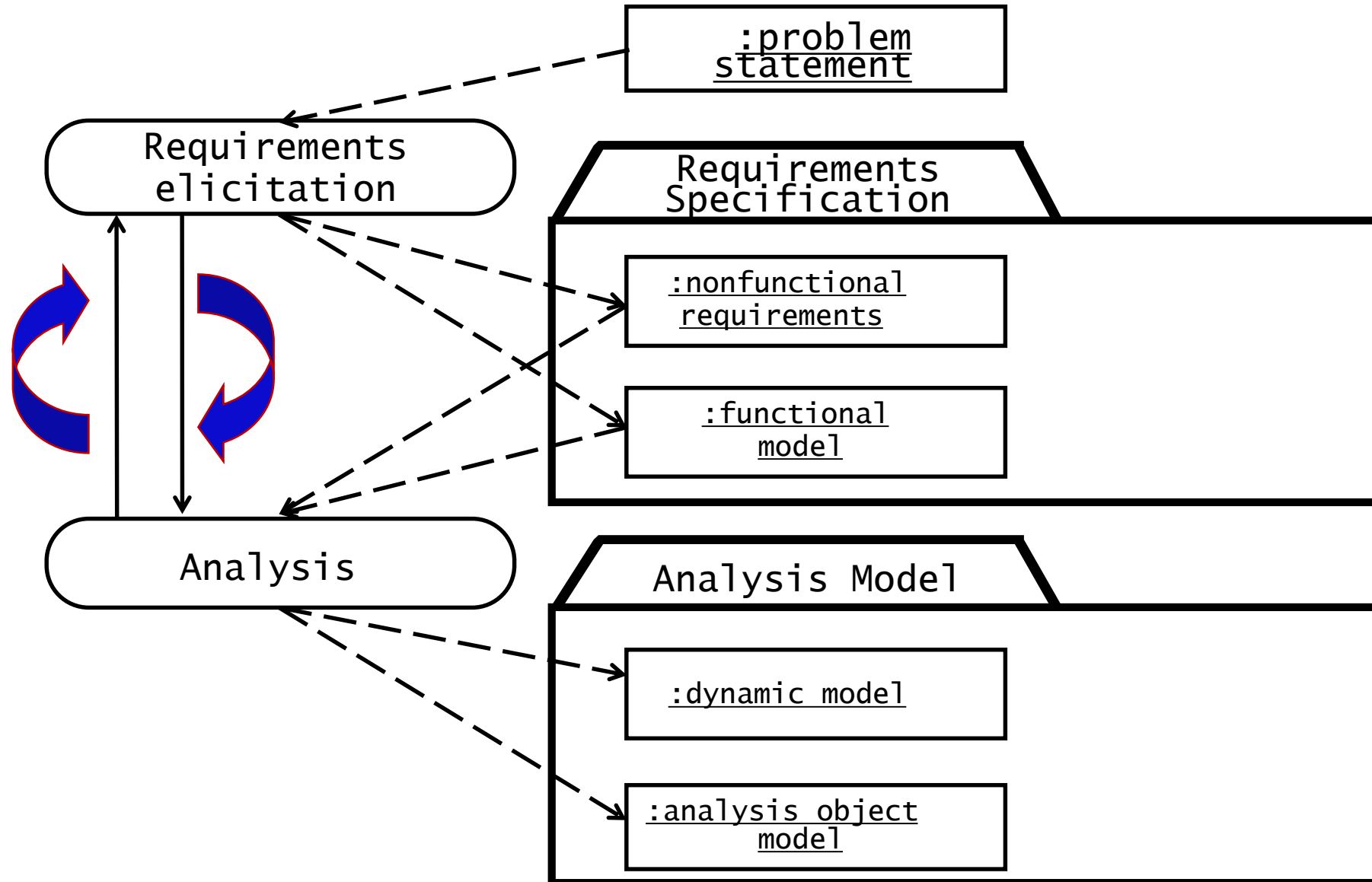
The laser beam was supposed to be reflected back towards a mountain top 10,023 feet high.

The operator entered the elevation as “10023”.

The light beam never hit the mountain top
What was the problem?

The computer interpreted the number in miles...

Requirements Process



Requirements Specification vs. Analysis Model

Both focus on the requirements from the user's view of the system

- The **requirements specification** uses natural language (derived from the problem statement)
- The **analysis model** uses a formal or semi-formal notation
 - We use UML