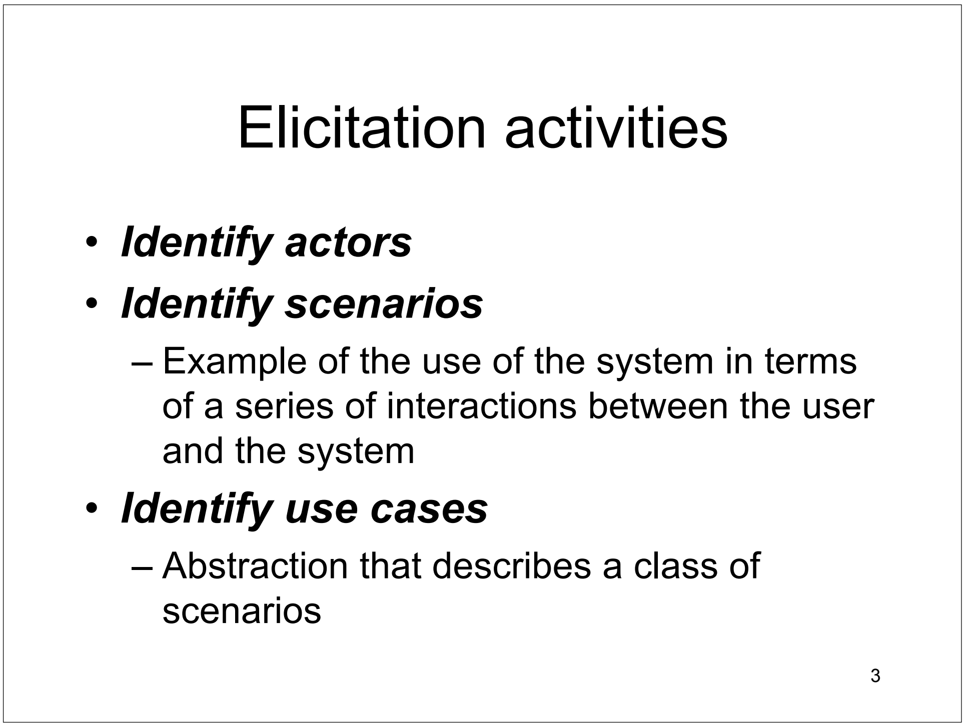
Requirements Engineering

How to do things in practice

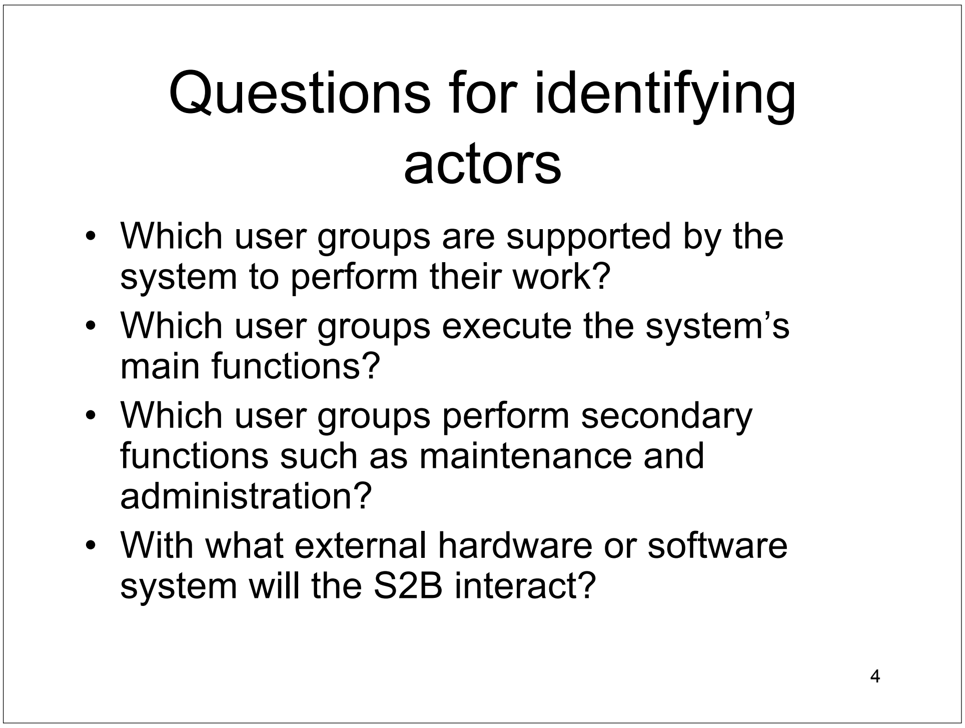
How to manage the process

- Requirements elicitation
  - the activities that leads to identifying *requirements*
- Analysis
  - the activities that derive a *specification*



Elicitation activities

- ***Identify actors***
- ***Identify scenarios***
  - Example of the use of the system in terms of a series of interactions between the user and the system
- ***Identify use cases***
  - Abstraction that describes a class of scenarios



Questions for identifying actors

- Which user groups are supported by the system to perform their work?
- Which user groups execute the system’s main functions?
- Which user groups perform secondary functions such as maintenance and administration?
- With what external hardware or software system will the S2B interact?

Scenarios

- A narrative description of what people do and experience as they try to make use of computer systems and applications
- A concrete, focused, informal description of a single feature of the system used by a single actor

Types of scenarios

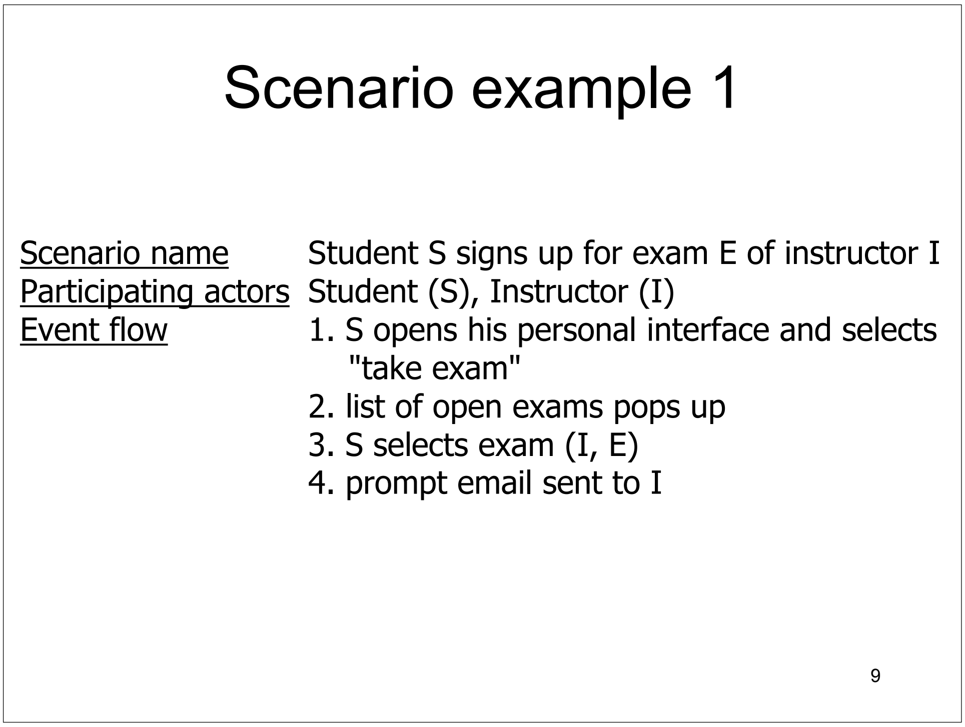
- As-is scenario
  - used in describing a current situation
  - derived from direct observation of user activities.
    - can be validated for correctness and accuracy with the users
- Visionary scenario
  - Used to describe future system functionalities

Finding scenarios: heuristics

- Ask the following questions:
  - What primary tasks will the system perform?
  - What data will the actors 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 *questionnaires* alone
- Insist on *task observation* if a system already exists (interface engineering or reengineering)
  - Speak to the end user, not just to the contractor
  - Expect resistance and try to overcome it

Exam management system

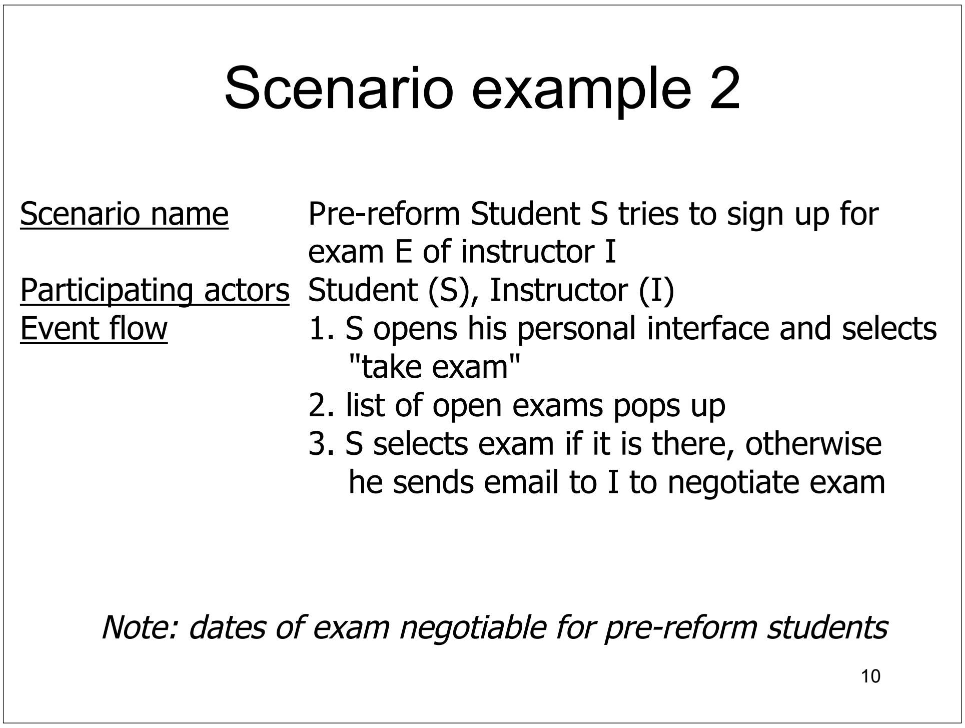
- What needs to be done to take SE exam?
- Who is involved?
- What exceptions can occur?



Scenario example 1

Scenario name	Student S signs up for exam E of instructor I
Participating actors	Student (S), Instructor (I)
Event flow	1. S opens his personal interface and selects "take exam" 2. list of open exams pops up 3. S selects exam (I, E) 4. prompt email sent to I

9



Scenario example 2

Scenario name	Pre-reform Student S tries to sign up for exam E of instructor I
Participating actors	Student (S), Instructor (I)
Event flow	1. S opens his personal interface and selects "take exam" 2. list of open exams pops up 3. S selects exam if it is there, otherwise he sends email to I to negotiate exam

Note: dates of exam negotiable for pre-reform students

10

Scenario example 3

Scenario name	Instructor I sets dates for exam E taken by pre-reform student S
Participating actors	Student (S), Instructor (I)
Event flow	1. I receives email from S asking for a date 2. I opens his personal interface and selects "pre-reform exam management" 3. a form pops up, with a list of courses 4. associate one or more dates for E

Another axample: Accident Management System

- What needs to be done to report a “Cat in a Tree” incident?
- What do you need to do if a person reports “Warehouse on Fire?”
- Who is involved in reporting an incident?
- What does the system do, if no police cars are available? If the police car has an accident on the way to the “cat in a tree” incident?
- What do you need to do if the “Cat in the Tree” turns into a “Grandma has fallen from the Ladder”?
- Can the system cope with a simultaneous incident report “Warehouse on Fire?”

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 her car.
- Alice enters the address of the building, a brief description of its location (i.e., north west corner), and an emergency level. In addition to a fire unit, she requests several paramedic units on the scene given that area appear to be relatively busy. 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 two paramedic units to the Incident site and sends their estimated arrival time (ETA) to Alice.
- Alice received the acknowledgment and the ETA.

Observations about Warehouse on Fire Scenario

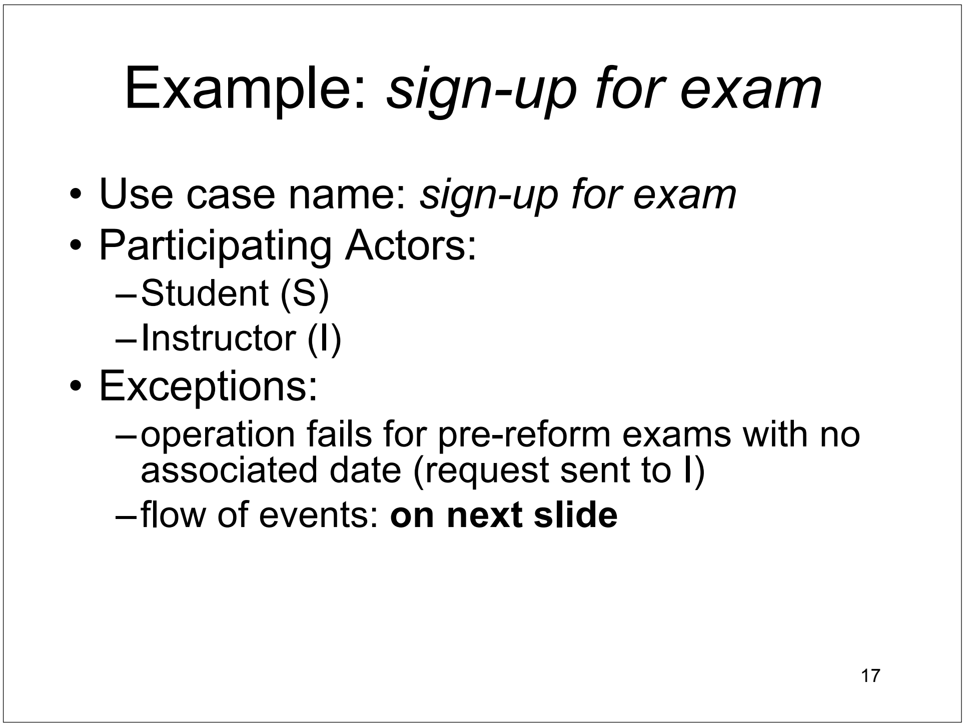
- Concrete scenario
  - Describes a single instance of reporting a fire incident.
  - Does not describe all possible situations in which a fire can be reported.
- Participating actors
  - Bob, Alice and John

Next goal, after the scenarios are formulated

- Find all the use cases in the scenario that specify all possible 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 Special Requirements (Constraints, Nonfunctional Requirements

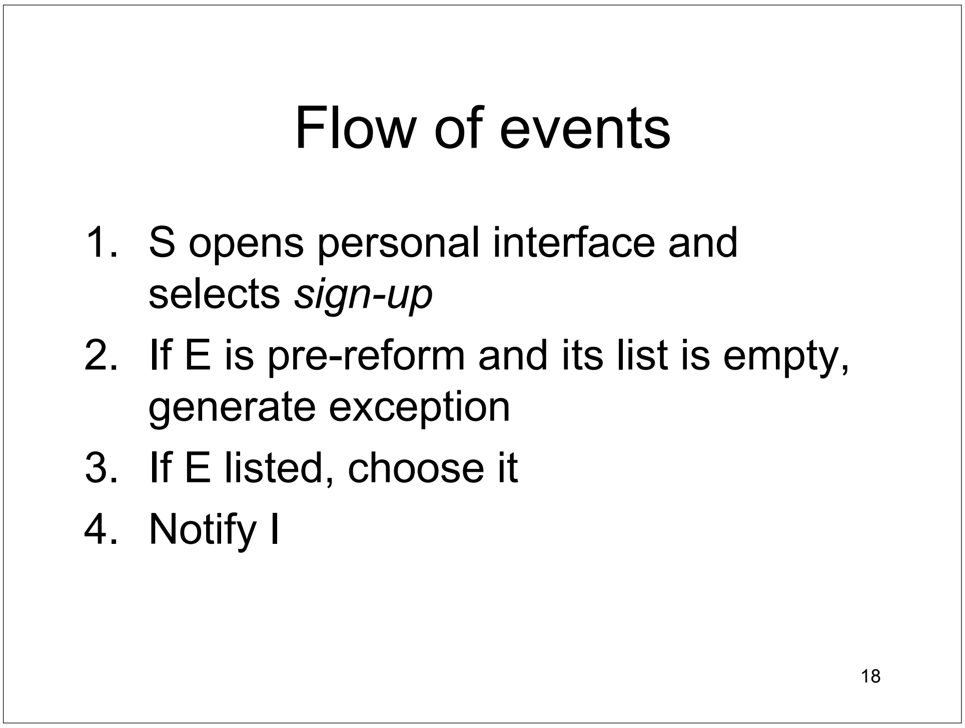
Use cases

- Scenarios lead to the identification of use cases
- Use cases represent the main functionalities required by the stakeholders
  – flow of events in the system, including interaction with actors
  – initiated by an actor
  – have a name
  – have a termination condition



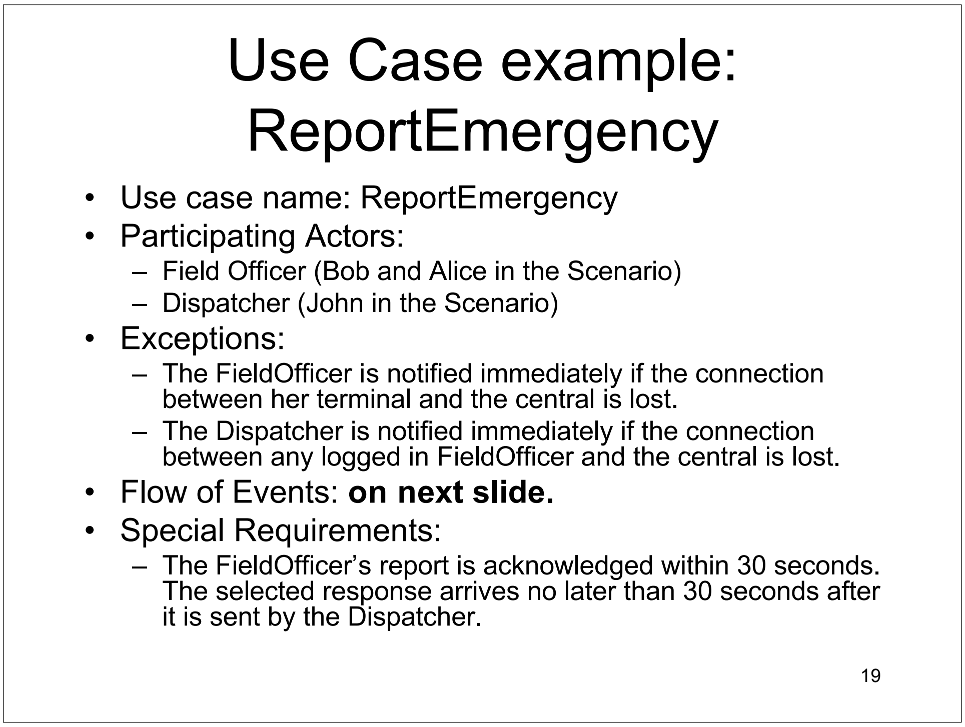
Example: *sign-up for exam*

- Use case name: *sign-up for exam*
- Participating Actors:
  - Student (S)
  - Instructor (I)
- Exceptions:
  - operation fails for pre-reform exams with no associated date (request sent to I)
  - flow of events: **on next slide**



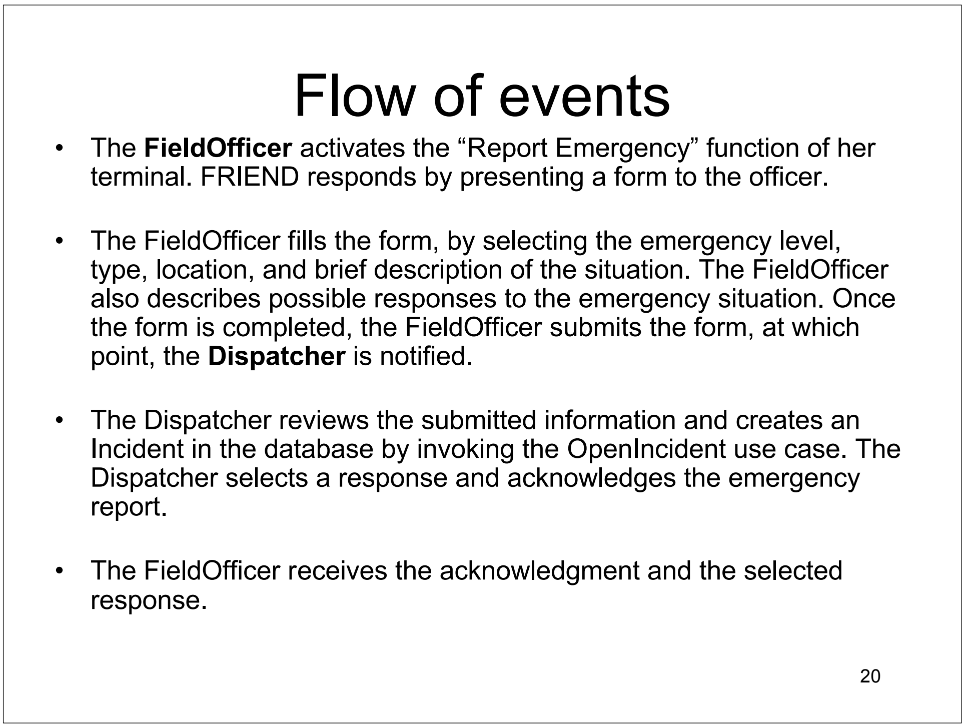
Flow of events

1. S opens personal interface and selects *sign-up*
2. If E is pre-reform and its list is empty, generate exception
3. If E listed, choose it
4. Notify I



Use Case example: ReportEmergency

- Use case name: ReportEmergency
- Participating Actors:
  - Field Officer (Bob and Alice in the Scenario)
  - Dispatcher (John in the Scenario)
- Exceptions:
  - The FieldOfficer is notified immediately if the connection between her terminal and the central is lost.
  - The Dispatcher is notified immediately if the connection between any logged in FieldOfficer and the central is lost.
- Flow of Events: **on next slide.**
- Special Requirements:
  - The FieldOfficer’s report is acknowledged within 30 seconds. The selected response arrives no later than 30 seconds after it is sent by the Dispatcher.



Flow of events

- The **FieldOfficer** activates the “Report Emergency” function of her terminal. FRIEND responds by presenting a form to the officer.

- The FieldOfficer fills the form, by selecting the emergency level, type, location, and brief description of the situation. The FieldOfficer also describes possible responses to the emergency situation. Once the form is completed, the FieldOfficer submits the form, at which point, the **Dispatcher** is notified.

- The Dispatcher reviews the submitted information and creates an Incident in the database by invoking the OpenIncident use case. The Dispatcher selects a response and acknowledges the emergency report.

- The FieldOfficer receives the acknowledgment and the selected response.

Tips to define proper use cases

- Use cases named with verbs that indicate what the user is trying to accomplish
- Actors named with nouns
- The boundary of the system should be clear, the steps accomplished by actors and those accomplished by the system should be clearly distinguished
- Use cases steps in active voice
- Causal relationship between steps should be clear
- A use case per user transaction
- Separate description of exceptions
- Keep use cases small (no more than two/three pages)

Example of a poor use case

- Use case name: Accident (not a verb)
- Participating Actors:
  - Field Officer (Dispatcher actor is missing here but mentioned in the next section)
- Flow of Events:
  - 1. The field officer report the accident
  - 2. An ambulance is dispatched (by who?)
  - 3. The Dispatcher is notified when the ambulance arrives on site (what does the field officer do after dispatching)?

Another Use Case example: allocate a resource

- Actors:
  - ***Field Supervisor:*** This is the official at the emergency site...
  - ***Resource Allocator:*** The Resource Allocator is responsible for the commitment and decommitment of the Resources managed by the FRIEND system. ...
  - ***Dispatcher:*** A Dispatcher enters, updates, and removes Emergency Incidents, Actions, and Requests in the system. The Dispatcher also closes Emergency Incidents.
  - ***Field Officer:*** Reports accidents from the Field

Another Use Case example: allocate a resource

- Use case name: AllocateResources
- Participating Actors:
  - Field Officer (Bob and Alice in the Scenario)
  - Dispatcher (John in the Scenario)
  - Resource Allocator
  - Field Supervisor
- Entry Condition
  - The Resource Allocator has selected an available resource.
  - The resource is currently not allocated
- Flow of Events
  - The Resource Allocator selects an Emergency Incident.
  - The Resource is committed to the Emergency Incident.
- Exit Condition
  - The use case terminates when the resource is committed.
  - The selected Resource is now unavailable to any other Emergency Incidents or Resource Requests.
- Special Requirements
  - The Field Supervisor is responsible for managing the Resources

Order of steps when formulating use cases

- First step: name the use case
  - Use case name: ReportEmergency
- Second step: Find the actors
  - Generalize the concrete names (“Bob”) to participating actors (“Field officer”)
  - Participating Actors:
    - Field Officer (Bob and Alice in the Scenario)
    - Dispatcher (John in the Scenario)
- Third step: Then concentrate on the flow of events
  - Use informal natural language

How to specify a Use Case (Summary)

- Name of Use Case
- Actors
  - Description of Actors involved in use case)
- Entry condition
  - “This use case starts when…”
- Flow of Events
  - Free form, informal natural language
- Exit condition
  - “This use cases terminates when…”
- Exceptions
  - Describe what happens if things go wrong
- Special Requirements
  - Nonfunctional Requirements, Constraints)

Analysis activities

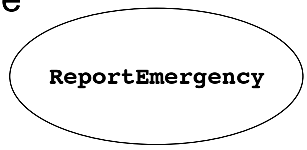
- Provide UML models
  - formalize use cases as UML use cases
  - formalize main entities as class diagrams
  - map use cases to objects with sequence diagrams
  - model state dependent behaviors with statecharts

UML will be taken up soon!

Hereafter a few hints on use cases

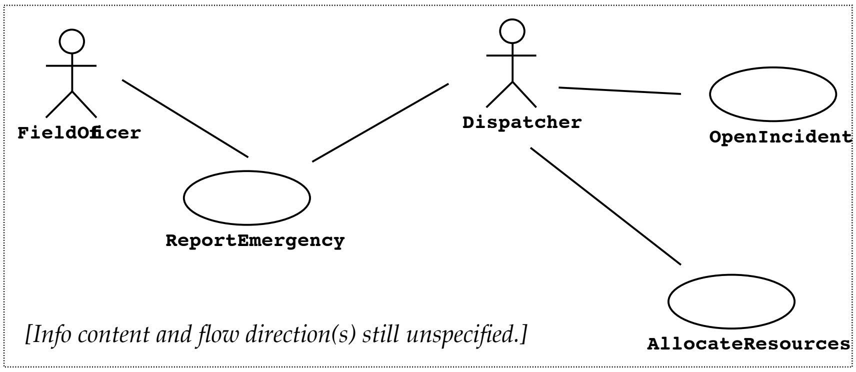
Use cases

- A use case is a flow of events in the system, including interaction with actors
- It is initiated by an actor
- Each use case has a name
- Each use case has a termination condition
- Graphical Notation: An oval with the name of the use case



Use Case Model: **The set of all use cases specifying the complete functionality of the system**

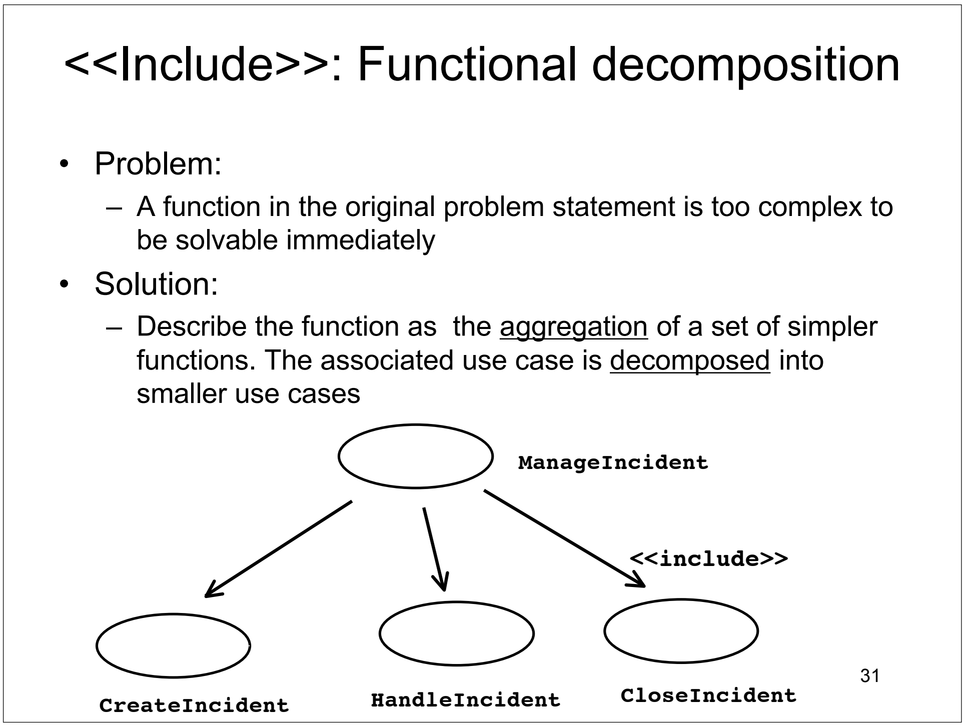
Example: Incident management



[Info content and flow direction(s) still unspecified.]

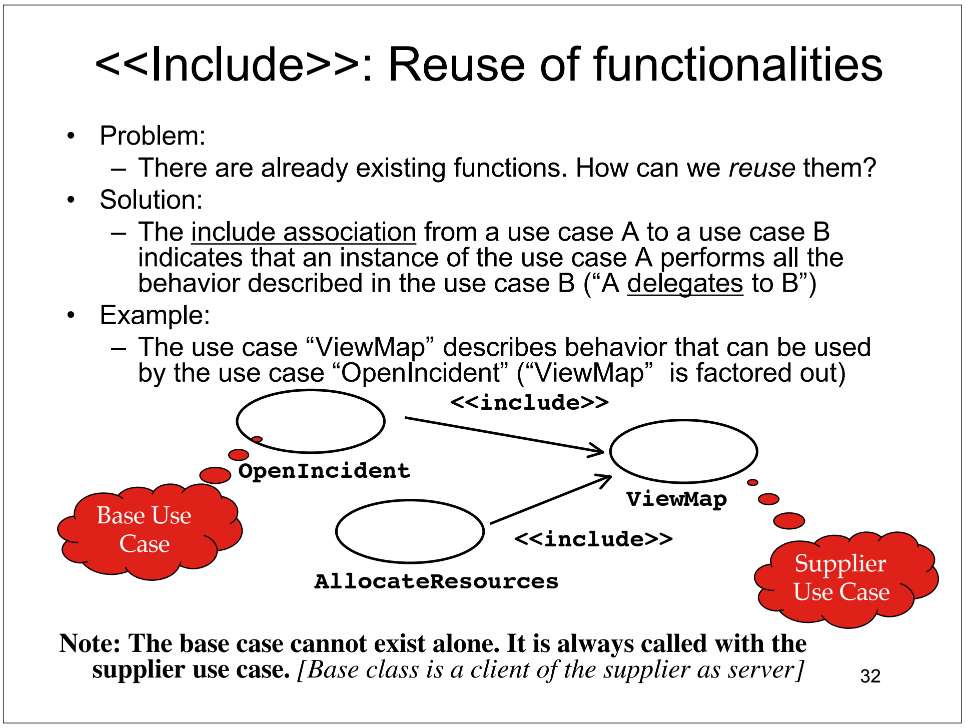
Use case associations

- A use case model consists of use cases and use case associations
  - A use case association is a relationship between use cases
- Important types of use case associations: Include, Extends, Generalization
- Include
  - A use case uses another use case (“functional decomposition”)
- Extends
  - A use case extends another use case
- Generalization
  - An abstract use case has several different specializations



<<Include>>: Functional decomposition

- Problem:
  - A function in the original problem statement is too complex to be solvable immediately
- Solution:
  - Describe the function as the aggregation of a set of simpler functions. The associated use case is decomposed into smaller use cases



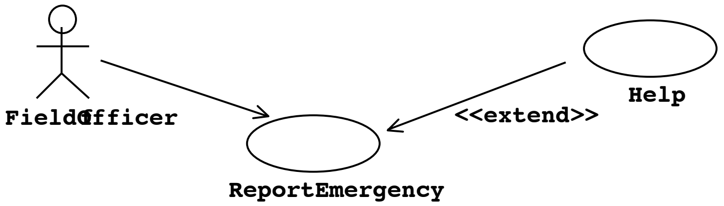
<<Include>>: Reuse of functionalities

- Problem:
  - There are already existing functions. How can we *reuse* them?
- Solution:
  - The include association from a use case A to a use case B indicates that an instance of the use case A performs all the behavior described in the use case B (“A delegates to B”)
- Example:
  - The use case “ViewMap” describes behavior that can be used by the use case “OpenIncident” (“ViewMap” is factored out)

Note: The base case cannot exist alone. It is always called with the supplier use case. *[Base class is a client of the supplier as server]*

<Extend>> association

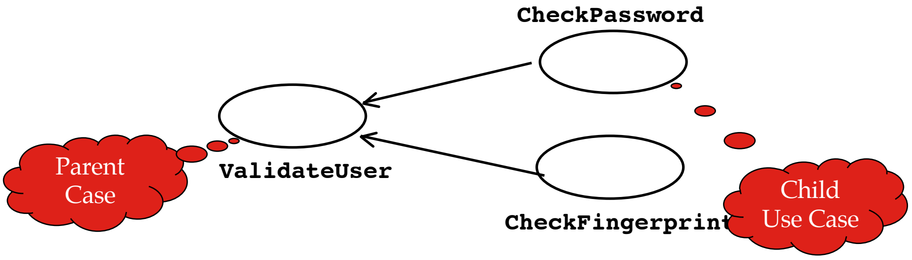
- Problem:
  - The functionality in the original problem statement needs to be extended.
- Solution:
  - An *extend association* from a use case A to a use case B indicates that use case B is an extension of use case A.
- Example:
  - The ***[base]*** use case “ReportEmergency” is complete by itself , but can be extended by the use case “Help” for a specific scenario in which the user requires help

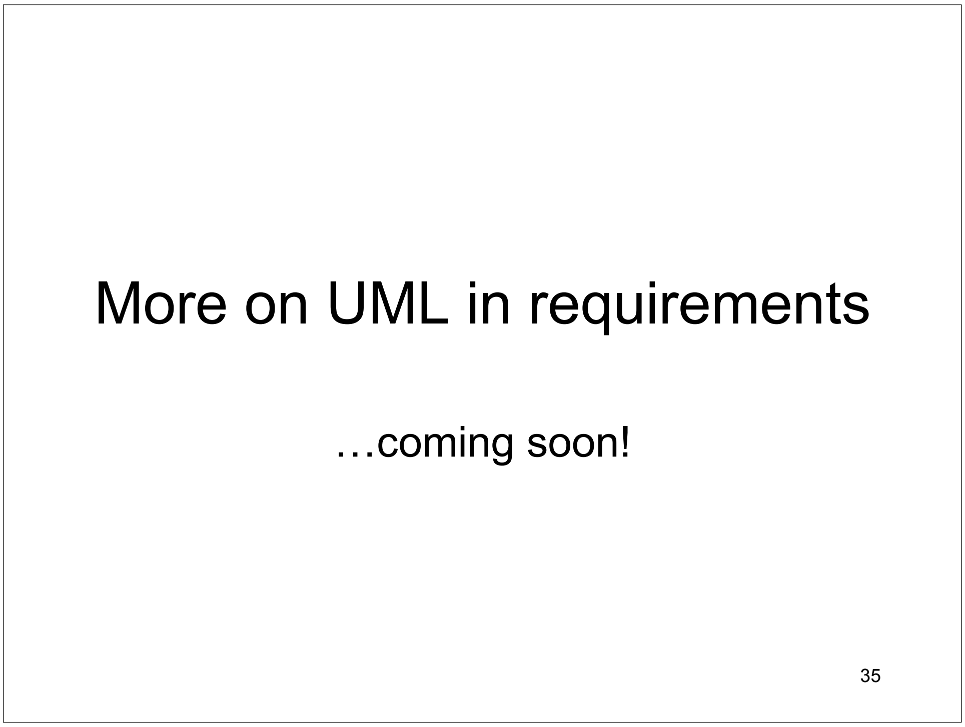


Note: The base use case can be executed without the use case extension in extend associations.

Generalization association

- **Problem**:
  - You have common behavior among use cases and want to factor this out.
- **Solution**:
  - The generalization association among use cases factors out common behavior. The child use cases inherit the behavior and meaning of the parent use case and add or override some behavior.
- **Example**:
  - Consider the use case “ValidateUser”, responsible for verifying the identity of the user. The customer might require ***[one of]*** two realizations: “CheckPassword” or “CheckFingerprint” ***[depending on the context].***





More on UML in requirements

…coming soon!