POLITECNICO
MILANO 1863

Software Engineering 2

Short summary about use cases

Relationship between use cases and requirements

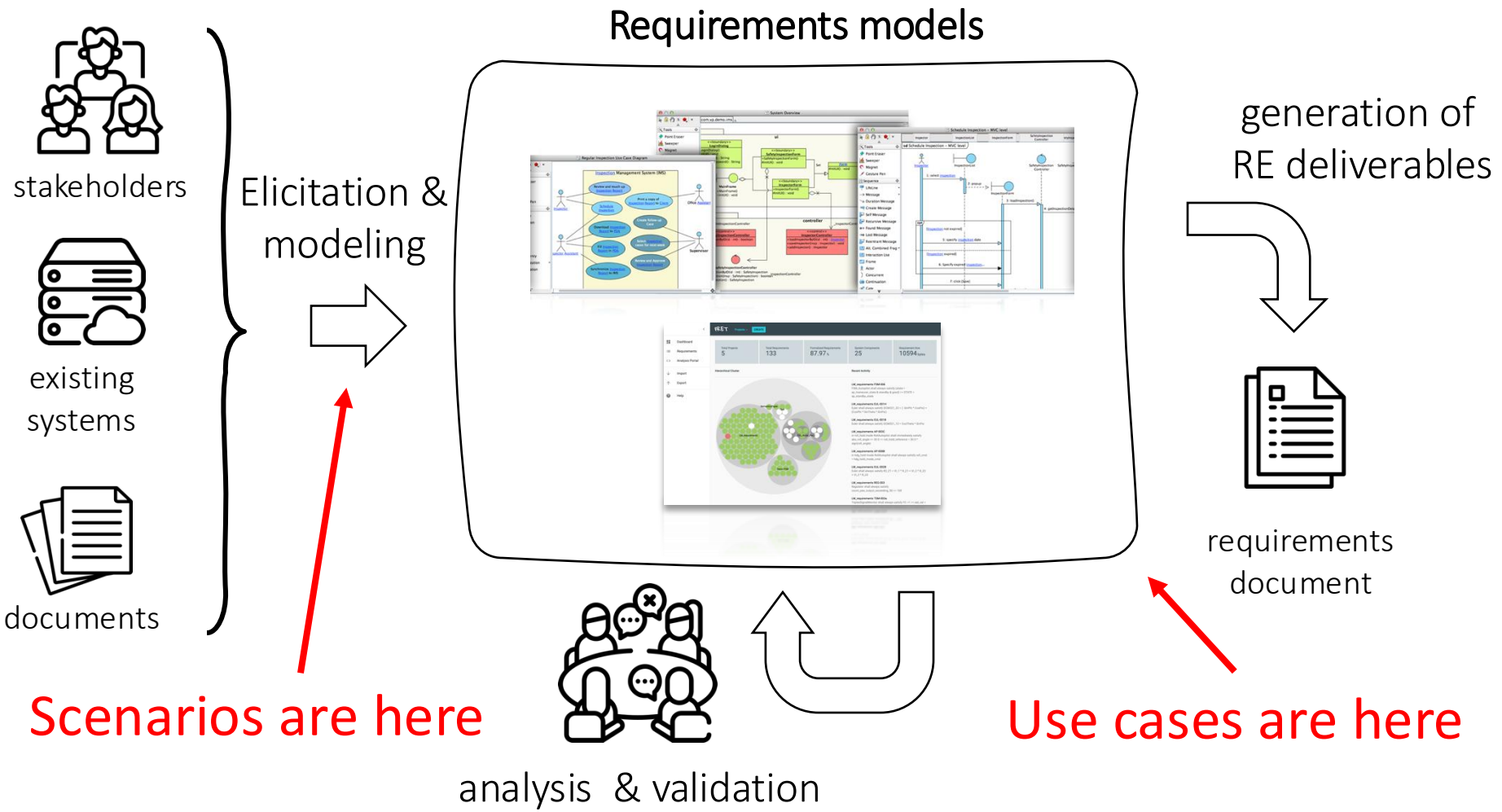
M Camilli, E Di Nitto, M Rossi

POLITECNICO
MILANO 1863

Requirements Engineering (RE)

Use cases: a summary

RE workflow: Requirements elicitation



Use case example: ReportEmergency (1)

POLITECNICO MILANO 1863

- **Participating actors**: FieldOfficer, Dispatcher
- **Entry Condition**: True (an emergency can be reported all times)
- **Flow of events**:
  - The **FieldOfficer** activates the “Report Emergency” function of her terminal.
  - **FRIEND** (the system to be developed) 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 allocates resources by invoking the AllocateResources use case. The Dispatcher selects a response and acknowledges the emergency report.
- **Exit condition**: The FieldOfficer has received the acknowledgment and the selected response.

Use case example: ReportEmergency (2)

- **Exceptions**:
  - The FieldOfficer is notified immediately if the connection between her terminal and the control room is lost
  - The Dispatcher is notified immediately if the connection between any logged in FieldOfficer and the control room is lost
- **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

Steps when formulating use cases

POLITECNICO MILANO 1863

- **1st step** → name the use case
  - Use case name: ReportEmergency
- **2nd step** → find the actors
  - Generalize the concrete names (“Bob”) to participating actors (“Field officer”)
- **3rd step** → concentrate on the flow of events, entry and exit conditions
  - Use informal natural language
- **4th step** → focus on exceptional cases and special requirements

How to specify a use case (summary)

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

Use cases and requirements

POLITECNICO MILANO 1863

- Each use case may lead to one or more requirements
- Examples from **ReportEmergency** and **AllocateResources**
  - FRIEND shall support FieldOfficers in reporting an emergency
  - FRIEND must have a response time lower than 30 seconds when reacting to emergency-related requests
  - FRIEND shall support Dispatchers in allocating the resources to the incident

- In turn, from the requirements, new, more detailed use cases could be derived describing how the requirements are fulfilled

POLITECNICO
MILANO 1863

Requirements Engineering (RE)

Examples of scenarios and use cases

Scenario on scheduling appointments (1/2)

POLITECNICO MILANO 1863

- Luigi has just received a phone call from his boss informing him that a meeting has been planned for the following day, from 1:00pm to 3:00pm
- His current schedule for that day is:
  - 8:00am-12:30pm: Working on the project (FixedTimeEvent: Working Event) (held at home)
  - 1:30pm-2:30pm: Lunch time (FlexibleTimeEvent: Personal Event, minimum length: 30 min) (held anywhere)

Scenario on scheduling appointments (2/2)

POLITECNICO MILANO 1863

- Luigi opens his mobile application, logs-in and inserts this new event from the homepage
  - He selects the type of the event choosing "working" and then "meeting" from the list
  - He adds a brief description of the meeting: “Meeting with Boss”
  - He then adds 1:00pm as starting time and 3:00pm as ending time, setting “home” as the starting location from which he will go to the meeting and "Harrison street" as location for the event
- The system calculates the possible paths from home to Harrison street, shows them to the user and confirms that there is enough time to get to the meeting place on time
- Since lunch time overlaps with this new event the system also throws a warning stating:
  - “Lunch time” overlaps with “Meeting with Boss”

Use case Insert Fixed Time Event (1/2)

Name	Insert Fixed Time Event
Actor	User, GPS Service
Entry condition	The User knows everything about the meeting he wants to plan.
Event Flow	1. In the homepage, the User clicks on the "New Event" button entering in the event creation page. 2. The User selects the type of event, by choosing it in a tree-like structured list, deciding to select as a top level of hierarchy, a working, personal or customized event. 3. The User gives a brief description of what the event will be about. 4. The User inserts the starting time, up to minutes precision, from a list box in order to avoid time not in the interval [00:00 - 23:59]. 5. The User inserts the ending time with the same procedure as the previous one, leaving the "flexible option" unchecked since he wants to insert a fixed time event. 6. The User inserts the starting position by giving the address or sharing the current position with the system using GPS technology. 7. The User inserts the position in which the event will take place. 8. The User clicks on the "Confirm" button 9. Included use case: Calculate travel paths 10. The system places the event in the calendar together with the information about how to reach it from the starting location

Use case Insert Fixed Time Event (2/2)

Exit condition	The event is in the User timetable. If there are overlaps with other events, they will be shown as warnings.
Exception	The system identifies an overlap with an event that is already in the calendar (in computing overlaps the system takes into account also the time needed to move from one location to the other) and shows them as warnings to the User offering the possibility to modify the schedule. User cannot find the event type he is looking for. In this case the User is able to insert a new type of event, exploiting the "Insert new type of event" functionality and than back to step one on the flow of event
Special Reqs	The daily timetable must be updated in less than 1 second.

Some World and Machine phenomena

POLITECNICO MILANO 1863

- **World-only phenomena**
  - User sets up meeting
  - User goes to meeting
- **Shared phenomena, world-controlled**
  - User inserts time of meeting
  - User inserts place of meeting
  - User selects type of event
  - User confirms meeting
- **Shared phenomena, machine-controlled**
  - System warns user of overlap between events
- **Machine-only phenomena**
  - System computes path
  - System stores event

Some goals and domain assumptions

POLITECNICO MILANO 1863

- **Goals**
  - G1: Users keep track of events in which they are involved
  - G2: Users avoid scheduling overlapping events
- **Domain Assumptions**
  - D1: The information about events inserted by users is correct
  - D2: GPS data is accurate (error of 20m at most)
- **Requirements**
  - R1: The system shall allow the user to create a new event
  - R2: The system shall allow the user to set the time of the event
  - R3: The system shall allow the user to select the type of the event
  - R4: The system shall allow the user to set the place of the event
  - R5: When user confirms an event, the system shall update the timetable in less than 1s

Relating the ingredients of Req elicitation and modeling