Where do we start?

• Identify the problem
  – what is the objective of the project?
  – the “vision” of those who are pushing for it?
    • e.g., “Meeting scheduling is too costly right now”

• Scope the problem
  – given the vision, how much do we tackle?
    • e.g. “Build a system that schedules meetings”, …or…
    • e.g. “Build a system that maintains people’s calendars” …or…

• Identify solution scenarios
  – given the problem, what is the appropriate business process for solving it?
    • e.g. “Anyone who wants to schedule a meeting goes to the secretary, gives details and the secretary handles the rest”, …or…

• Scope the solution
  – Given a business process, what parts should be automated, and how?
    • e.g. “Computer takes in scheduling request details, outputs a solution” …or…
    • e.g. “Solution arrived at interactively by secretary and computer” …or…
Identifying the Problem

• Vague problem stated by the customer:
  – E.g. university textbook store:
    • Manager wants to computerize the book order forms filled out by instructors;
  – E.g. A large insurance company:
    • Claims manager wants to cut down the average time it takes to process an insurance claim from 2 months to 2 weeks
  – E.g. A telecommunications company:
    • CIO wants to integrate the billing system with customer record systems of several affiliates, so there is only one billing system...
  – E.g. Large Government Aerospace Agency:
    • The president wants to send a manned mission to Mars by the year 2020

• Often you only see symptoms rather than causes:
  – E.g. “Ontario patients needing X-ray scans have to wait for months”

• The long wait is the symptom, not the problem. The problem may be:
  • Shortage of X-ray machines;
  • Shortage of trained staff;
  • Shortage of doctors to process the data
  • Inefficient scheduling procedures
Lecture 2:

Stakeholders, Goals and Scenarios

• Stakeholders
  – Identifying the problem owners

• Goals
  – Identifying the success criteria

• Scenarios
  – Identifying how it works
Stakeholders

- **Stakeholder analysis:** Identify all the people who must be consulted during information acquisition.

- **Example stakeholders**
  - Users
    - concerned with the features and functionality of the new system
  - Designers
    - want to build a perfect system, or reuse existing code
  - Systems analysts
    - want to “get the requirements right”
  - Training and user support staff
    - want to make sure the new system is usable and manageable
  - Business analysts
    - want to make sure “we are doing better than the competition”
  - Technical authors
    - will prepare user manuals and other documentation for the new system
  - The project manager
    - wants to complete the project on time, within budget, with all objectives met.
  - “The customer”
    - Wants to get best value for money invested!

Levels of authority

- **Top management**
  - establishes goals
  - does long-range planning
  - determines new market & product developments
  - decides on mergers & acquisitions.

- **Middle management**
  - sets objectives
  - allocates & controls resources
  - does planning
  - measures performance

- **Lower management**
  - supervises day-to-day operations
  - takes corrective action when necessary.

- **Operational level**
  - performs day-to-day operations
Lecture 2: Stakeholders, Goals and Scenarios

• **Stakeholders**
  – Identifying the problem owners

• **Goals**
  – Identifying the success criteria

• **Scenarios**
  – Identifying how it works

---

**Goals**

• **Approach**
  – Focus on *why* a system is required
  – Use goal refinement to arrive at specific requirements
  – Goal analysis
    • document, organize and classify goals
  – Goal hierarchies show refinements and alternatives

• **Advantages**
  – Reasonably intuitive
  – Explicit declaration of goals provides sound basis for conflict resolution

• **Disadvantages**
  – Captures a static picture - what if goals change over time?
  – Can regress forever up (or down) the goal hierarchy

• **Goals:**
  – Describe functions that must be carried out

• **Actors:**
  – Owners of goals

• **Tips:**
  – Multiple sources - better goals
  – Associate stakeholders with each goal
  – Use scenarios to explore how goals can be met
Goal Modeling

• **(Hard) Goals:**
  - Describe functions that must be carried out. E.g.
    • Satisfaction goals
    • Information goals

• **Softgoals:**
  - Cannot really be fully satisfied. E.g.
    • Accuracy
    • Performance
    • Security
    • ...

• **Also classified temporally:**
  - Achieve/ Cease goals
    • Reach some desired state eventually
  - Maintain/ Avoid goals
    • Keep some property invariant
  - Optimize
    • A criterion for selecting behaviours

• **Agents:**
  - Owners of goals
  - Choice of when to ascribe goals to agents:
    • Identify agents first, and then their goals
    • Identify goals first, and then allocate them to agents during operationalization

• **Modelling Tips:**
  - Multiple sources yield better goals
  - Associate stakeholders with each goal
    • reveals viewpoints and conflict
  - Use scenarios to explore how goals can be met
  - Explicit consideration of obstacles helps to elicit exceptions

Example Goal Elaboration

Or-decomposition

Crucial planning decision be made

Decision be made by email discussion

Decision be made face-to-face

Meeting be held

Minutes be circulated

Agenda be defined

Meeting be scheduled

Date and location set

Attendees know details

Changes be handled

Meeting be requested

Attendee list obtained

AV & other needs defined

Attendees’ preferences known

Room availability determined

Meeting announced

Facilities booked

Attendance confirmed

Participants notified

Change requests accepted
Goal analysis

• **Relationships between goals:**
  – One goal **helps** achieve another (+)
  – One goal **hurts** achievement of another (-)
  – One goal **makes** another (++)
    • Achievement of goal A guarantees achievement of goal B
  – One goal **breaks** another (--)  
    • Achievement of goal A prevents achievement of goal B

• **Goal Elaboration:**
  – “**Why**” questions explore higher goals (context)
  – “**How**” questions explore lower goals (operations)
  – “**How else**” questions explore alternatives

Softgoals

• **Some goals can never be fully satisfied**
  – Treat these as **softgoals**
    • E.g. “system be easy to use”; “access be secure”
    • Also known as ‘non-functional requirements’; ‘quality requirements’
  – Will look for things that contribute to **satisficing** the softgoals
  – E.g. for a train system:
Softgoals as selection criteria

- **Strategic dependency model**
  - used to express the network of intentional, strategic relationships among actors

- **Strategic rationale model**
  - used to express the rationales behind dependencies

http://istar.rwth-aachen.de/
Strategic dependency model (1)

- **Actor**
  - carries out actions to achieve goals
- **Role**
  - characterization of the behavior of a social actor within some context
  - a set of *roles* typically played by one *agent*
- **Agent**
  - actor with concrete, physical manifestations, such as a human individual
  - an *agent* occupies a *position*
- **Position**
  - used between a role and an agent
  - a *position* is said to cover a *role*

Strategic dependency model (2)

- **Dependee**
  - Actor who is depended upon on a dependency relationship.
- **Depender**
  - The depending actor on a dependency relationship.
- **Dependum**
  - Element around which a dependency relationship centers.
Strategic dependency model (3)

- **Goal dependency**
  - the depender depends on the dependee to bring about a certain state of affairs in the world

- **Task dependency**
  - the depender depends on the dependee to carry out an activity

- **Resource dependency**
  - the depender depends on the dependee for the availability of an entity

- **Softgoal dependency**
  - a depender depends on the dependee to perform some task that meets a softgoal

Strategic dependency model (4)
Strategic rationale model (1)

- **Actor boundaries**
  - all of the elements within a boundary for an actor are explicitly desired by that actor
  - to achieve these elements, an actor must depend on the intentions of other actors

- **Goal (hardgoal)**
  - intentional desire of an actor

- **Softgoal**
  - criteria for the goal's satisfaction are not clear-cut
  - judged to be sufficiently satisfied from the point of view of the actor

- **Task**
  - actor wants to accomplish some specific task, performed in a particular way

- **Resource**
  - actor desires the provision of some entity, physical or informational

Strategic rationale model (2)

- **Means-ends**
  - a relationship between an end, and a means for attaining it
  - "means" is expressed in the form of a task
  - "end" is expressed as a goal

- **Decomposition**
  - task can be decomposed into four types of elements: a subgoal, a subtask, a resource, and/or a softgoal
Strategic rationale model (3)

• Contribution
  – **Make**: strong enough to satisfice a softgoal
  – **Some+**: positive with unknown strength
  – **Help**: not sufficient by itself to satisfice the softgoal
  – **Unknown**: polarity is unknown
  – **Break**: sufficient enough to deny a softgoal
  – **Some-**: negative with unknown strength
  – **Hurt**: not sufficient by itself to deny the softgoal
  – **Or**: satisficed if any of the offspring are satisficed
  – **And**: satisficed if all of the offspring are satisficed
Lecture 2:

Stakeholders, Goals and Scenarios

• Stakeholders
  – Identifying the problem owners

• Goals
  – Identifying the success criteria

• Scenarios
  – Identifying how it works

Goals document stakeholder intentions

• Scenarios document sequences of interactions in which the system either satisfies some goals or fail to satisfy them

• Scenario classes:
  – Current state vs. Desired state scenarios
  – Positive vs. Negative scenarios
  – Misuse scenarios
  – Descriptive vs. Exploratory vs. Explanatory scenarios
  – Main scenario, Alternative scenario, Exceptional scenario
Narrative Scenario

- The driver assistance system includes a (sub-)system for avoiding rear-end collisions. This system comprises distance sensors that permanently check for distance to the vehicle driving ahead in order to avoid an imminent rear-end collision. In the system detects that the distance falls below the safety yet is still outside the critical range, an acoustic warning signal sounds. Alternatively, a symbol or message may be displayed on the driver display in the cockpit of the car. If the driver has not reacted to the warning after 2s and the distance between the two cars still decreasing, the system reduces the speed of the car. If the distance (in meters) falls below one quarter of the driving speed of the car (in km/h) at any time, the system initiates emergency braking.

- Static/structural aspect
- Functional aspect
- Explanatory aspect
- Behavioral aspect

Structured scenario

1. The driver activates the navigation system
2. The navigation system determines the current position of the car
3. The navigation system asks for the desired destination
4. The driver enters the destination
5. The navigation system identifies the relevant part of the map
6. The navigation system displays the map of the destination area
7. The navigation system asks for the routing options
8. The driver selects the desired routing options
9. The navigation system calculates the route
10. The navigation system informs the driver that the route has been calculated
11. The navigation system creates a list of waypoints
12. The navigation system displays the next waypoint of the calculated route
<table>
<thead>
<tr>
<th>Driver</th>
<th>Navigation system</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The driver activates the navigation system</td>
<td>2. The navigation system determines the current position of the car</td>
</tr>
<tr>
<td></td>
<td>3. The navigation system asks for the desired destination</td>
</tr>
<tr>
<td>4. The driver enters the destination</td>
<td>5. The navigation system identifies the relevant part of the map</td>
</tr>
<tr>
<td></td>
<td>6. The navigation system displays the map of the destination area</td>
</tr>
<tr>
<td></td>
<td>7. The navigation system asks for the routing options</td>
</tr>
<tr>
<td>8. The driver selects the desired routing options</td>
<td>9. The navigation system calculates the route</td>
</tr>
<tr>
<td></td>
<td>10. The navigation system informs the driver that the route has been calculated</td>
</tr>
<tr>
<td></td>
<td>11. The navigation system creates a list of waypoints</td>
</tr>
<tr>
<td></td>
<td>12. The navigation system displays the next waypoint of the calculated route</td>
</tr>
</tbody>
</table>

Self-study

- Reference Template for Scenario
- Reference Template for use cases
- 11 rules for documenting scenarios
Benefits of using Goals

• **Documentation**
  – Checking requirements for completeness
  – Avoiding irrelevant requirements
  – Structuring of documents

• **Elicitation**
  – Foundation for requirements elicitation
  – Identification and evaluation of alternative realisations
  – Refinement of the vision

• **Negotiation**
  – Consolidation of different views
  – Supporting conflict resolution

• **Validation**
  – Validity of requirements with respect to goal model

• **Management**
  – Prioritisation of requirements
  – Traceability of requirements

Benefits of using Scenarios

• **Documentation**
  – Guiding requirements engineers by means of scenarios
  – Structuring requirements documents
  – Providing rich context information for requirements
  – Embedding requirements into a usage context

• **Elicitation**
  – Goal refinement
  – Explanation of intentions
  – Communication support
  – Basis for development of requirements

• **Negotiation**
  – Conflict analysis and resolution

• **Validation**
  – Validating scenarios from different perspectives
  – Detecting irrelevant requirements
  – Including context information in the validation

• **Management**
  – Prioritisation support of requirements
  – Determining required traceability information
  – Support of change management
Goal-Scenario coupling

... initiate and influence the definition of ...

... classify ...

Goals

... illustrate satisfaction...

... lead to the identification of new ...

Scenarios

... lead to revision of ...

Lecture 2:

Stakeholders, Goals and Scenarios

• Stakeholders
  – Identifying the problem owners

• Goals
  – Identifying the success criteria

• Scenarios
  – Identifying how it works
Use case templates (1)
(Wiegers, 2004)

| Use Case ID: |  |
| Use Case Name: |  |
| Created By: | Last Updated By: |
| Date Created: | Date Last Updated: |

| Actors: |  |
| Description: |  |
| Trigger: |  |
| Preconditions: | 1. |
| Postconditions: | 1. |
| Normal Flow: | 1. |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Priority: |  |
| Frequency of Use: |  |
| Business Rules: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

Use case templates (2)
(Wiegers, 2004)

| Use Case ID: |  |
| Use Case Name: |  |
| Created By: | Last Updated By: |
| Date Created: | Date Last Updated: |

- **Use Case ID**: a unique integer sequence number identifier
- **Use Case Name**: a concise, results-oriented name for the use case
- **Created By**: the name of the person who initially documented this use case
- **Date Created**: the date on which the use case was initially documented
- **Last Updated By**: the name of the person who performed the most recent update to the use case description
- **Date Last Updated**: the date on which the use case was most recently updated
### Use case templates (3)
(Wiegers, 2004)

<table>
<thead>
<tr>
<th>Use Case ID:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name:</td>
</tr>
<tr>
<td>Created By:</td>
</tr>
<tr>
<td>Date Created:</td>
</tr>
</tbody>
</table>

- **Actors**: a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks
- **Description**: the reason for and outcome of this use case, the sequence of actions and the outcome of executing the use case
- **Trigger**: the event that initiates the use case
- **Pre-condition**: list any activities that must take place, or any conditions that must be true, before the use case can be started
- **Post-condition**: the state of the system at the conclusion of the use case execution

### Use case templates (4)
(Wiegers, 2004)

<table>
<thead>
<tr>
<th>Use Case ID:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name:</td>
</tr>
<tr>
<td>Created By:</td>
</tr>
<tr>
<td>Date Created:</td>
</tr>
</tbody>
</table>

- **Normal flow**: a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions
- **Alternative flows**: other, legitimate usage scenarios that can take place
- **Exceptions**: any anticipated error conditions that could occur during execution of the use case
- **Includes**: any other use cases that are included (“called”) by this use case
- **Priority**: the relative priority of implementing the functionality required to allow this use case
## Use case templates

(Wiegers, 2004)

<table>
<thead>
<tr>
<th>Use Case ID:</th>
<th>Use Case Name:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Created By:</td>
<td>Last Updated By:</td>
</tr>
<tr>
<td>Date Created:</td>
<td>Date Last Updated:</td>
</tr>
</tbody>
</table>

- **Frequency of use**: the number of times this use case will be performed by the actors per some appropriate unit of time
- **Business rules**: any business rules that influence this use case
- **Special Requirements**: any additional requirements (e.g., quality) for the use case that may need to be addressed during design or implementation
- **Assumptions**: any assumptions that were made in the analysis that led to accepting this use case
- **Notes and Issues**: any additional comments

---

**Sequences of Events**

- **Sequence Diagrams**
  - Sequence diagrams show how objects interacts with one another
  - Sequence diagrams…
    - …should remain easy to read and understand
    - …do not include complex control logic
Example

Initiator: Person
Staff: Person
Scheduler: Person
Participant: Person

Call() → Respond()
What's up?() → Give mtg details()
[for all participants] *Inform()
 Acknowledge()
[for all participants] *Remind()
 Prompt()
Show schedule()
[decision=OK] ScheduleOK'ed()

Another example

Campaign Manager

getName() → :Client
listCampaigns() → *getCampaign Details()
listAdverts() → *getAdvert Details()
addNewAdvert() → newAd Advert

Object lifetime
Activation
Object creation