BREN:ICS
INFORMATION AND COMPUTER SCIENCES

Identifying User Needs and Establishing Requirements.

Interaction Design, Chapter 7

Tempe Kraus
Yongjie Zheng

October 30, 2007

Outline

- What are we trying to achieve?
  - Identifying needs and establishing requirements
  - Categories of requirements
- Data gathering techniques
  - Choosing between data gathering techniques
  - Data gathering guidelines
- Data interpretation and analysis
- Task description and analysis
  - Scenarios, use cases, essential use cases and task analysis
- Summary
- Additional References

In the beginning... What are we trying to achieve?

- Identifying needs:
  - Understand as much as possible about the users, as well as their work and the context of their work.
  - System under development should support users in achieving their goals.

- Identifying needs is crucial to our next step.

- Establishing requirements:
  - Building upon the needs identified, produce a set of requirements.

- A user-centered approached to development:
  - Study that investigated the causes of IT project failure found that “requirements definition” was the most frequently cited project stage that caused failure.
  - Understanding what the product should do and making sure it meets the stakeholders’ needs are absolutely critical to the success of the product.

What are requirements?

- A requirement is a statement that specifies what an intended product should do, or how it should perform.

- Traditionally, two types of requirements:
  - Functional requirements specify what the system should do.
  - Non-Functional requirements specify what constraints there are on the system or its development.

- Interaction design requires us to understand *both* the functionality required and the constraints for development or operation of the product.

- Let’s refine these two broad types into further categories.

Categories of requirements

Category	Description
• Functional requirements	• What the product should do.
• Data requirements	• The type, volatility, size/amount, persistence, accuracy and value of the amounts of the required data.
• Environmental requirements	• Or “context of use” – circumstances in which the interactive product must operate.
• User requirements	• Characteristics of the intended user group.
• Usability requirements	• The usability goals and associated measures.

Source: *Interaction Design*, ch. 7

Data gathering

Overview of data gathering techniques used in the requirements activity

Technique	Good for	Kind of data	Advantages	Disadvantages
Questionnaires	Answering specific questions	Quantitative and qualitative data	Can reach many people with low resource	Design is crucial and response rate may be low. Responses may not be useful.
Interviews	Exploring issues	Some quantitative but mostly qualitative data	Interviewer can guide interviewee if necessary. Encourages contact between developers and users.	Time consuming. Artificial environment may intimidate interviewee.
Focus groups and workshops	Collecting multiple viewpoints	Some quantitative but mostly qualitative data	Highlights areas of consensus and conflict. Encourages contact between developers and users.	Possibility of dominant characters
Naturalistic observation	Understanding context of user activity	Qualitative	Observing actual work gives insights that other techniques can't give	Very time consuming. Huge amounts of data.
Studying documentation	Learning about procedures, regulations and standards	Quantitative	No time commitment from users required	Day-to-day working will differ from documented procedures

Source: *Interaction Design*, ch. 7

Choosing between data gathering techniques

- Your choice is influenced by a number of factors.
- The kind of information you want.
  - May also change depending on the stage of the project.
- The resources available to you.
  - E.g., your project may not have the time, money or personnel to send out a nationwide survey.
- The location and accessibility of stakeholders.
  - You may want to run a workshop for a large group of stakeholders, but could be prohibited by geography.

Choosing between data gathering techniques, continued

- Two main issues to consider when making your choice:
  - The nature of the data gathering technique itself.
  - The task which is to be studied.

- Data gathering techniques differ in the following:
  - The amount of time they take, level of detail and risk associated with the findings.
  - The knowledge the analyst must have about basic cognitive processes.

- Tasks can be classified along three scales:
  - Is the task a set of sequential steps or is it a rapidly overlapping series of subtasks?
  - Does the task involve high information content with complex visual displays, or low information content, where simple signals are enough to alert the user?
  - Is the task intended to be performed by a laymen with minimal training, or a practitioner highly skilled in the task domain?

- Example: the design of an ATM vs. the design of a system to support back-room workers at a bank who are reconciling the machine register with the customers' deposit slip.

Basic data gathering guidelines

- Focus on identifying the stakeholders’ needs.
- Involve all the stakeholder groups.
- Involve more than one representative from each stakeholder group.
- Use a combination of data gathering techniques.
- Support the data-gathering sessions with suitable props.
- Run a pilot session if possible, to work out any kinks.
- Understand what you are really looking for (though compromise may be needed).
- Carefully consider the means used to record the data during a face-to-face data gathering session.

Data interpretation and analysis

- Once you have gathered your data, you will need to interpret and analyze it.
  - Start interpretation and analysis as soon after the gathering session as possible.

- Interpreting data:
  - Begin structuring and recording descriptions of requirements.
  - Capture information in documents and diagrams.
  - This helps to keep track of context and usage information during the rest of the process.

- Analyzing data:
  - Data-flow diagrams, state charts, work-flow charts, etc.
  - For object-oriented approaches, can use class diagrams, sequence diagrams, etc.

- Requirements activity iterates numerous times before stable requirements evolve.
  - Continued interpretation and analysis throughout the process will result in a deeper understanding as well as clarification of the requirements.

- We will focus on four techniques that have a user-centered focus and are intended to understand the users’ goals and tasks.

Task description and analysis

- User-centered task descriptions are created to understand users’ goals and tasks.
  - Scenarios
  - Use cases
  - Essential use cases
  - Task analysis

- Methodology for each:
  - Description
  - Advantages
  - Limitations
  - How to develop

- Example:
  - The shared calendar application

- System-centered descriptions are used to communicate precise information with developers.

Task description and analysis, continued

Scenarios

Description

Describes human activities or tasks in a story that allows exploration and discussion of contexts, needs, and requirements.

Advantages

Telling a story is a natural way for people to explain what they are doing or how to achieve something. It also allows us to identify the stakeholders and the products involved in the activity.

Limitations

More focused on task characteristics than the detail of interface design and layout. [2]

To Develop

Focus on what users are trying to achieve.

Task description and analysis, continued

- Scenarios: the shared calendar example

The user types in all the names of the meeting participants together with some constraints such as the length of the meeting, roughly when the meeting needs to take place, and possibly where it needs to take place. The system then checks against the individuals' calendars and the central department calendar and presents the user with a series of dates on which everyone is free all at the same time. Then the meeting could be confirmed and written into peoples' calendars. Some people, though, will want to be asked before the calendar entry is made. Perhaps the system could email them automatically and ask that it be confirmed before it is written in.

Source: *Interaction Design*, ch. 7

Task description and analysis, continued

Use Cases

Description

The main emphasis is on user-system interaction, but also user goals.

Advantages

It is easy to grasp key features in the user-system interaction activities.

Limitations

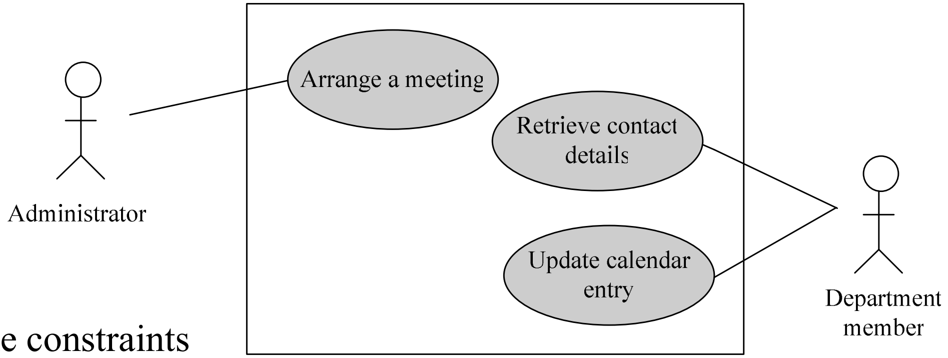
Traditional use cases contain certain assumptions, including that there is a piece of technology to interact with, and the kind of interaction to be designed.

To Develop

Identify the actors, then examine these actors and identify their goal or goals in using the system.

Task description and analysis, continued

- Use cases: the shared calendar example



1. The user chooses the option to arrange a meeting
2. The system prompts user for the names of attendees
3. The user types in a list of names
4. The system checks that the list is valid
5. The system prompts the user for meeting constraints
6. The user types in meeting constraints
7. The system searches the calendars for a date that satisfies the constraints
8. The system displays a list of potential dates
9. The user chooses one of the dates
10. The system writes the meeting into the calendar
11. The system emails all the meeting participants informing them for the appointment

Alternative courses:

5. If the list of people is invalid
   5.1 The system displays an error message
   5.2 The system returns to step 2
8. If no potential dates are found
   8.1 The system displays a suitable message
   8.2 The system returns to step 5

Source: *Interaction Design*, ch. 7

Task description and analysis, continued

Essential Use Cases

Description

A structured narrative consisting of three parts: a name that expresses the overall user intention, a stepped description of user actions, and a stepped description of system responsibilities.

Advantages

Represents a more general case than a scenario embodies, and tries to avoid the assumptions of a traditional use case.

Limitations

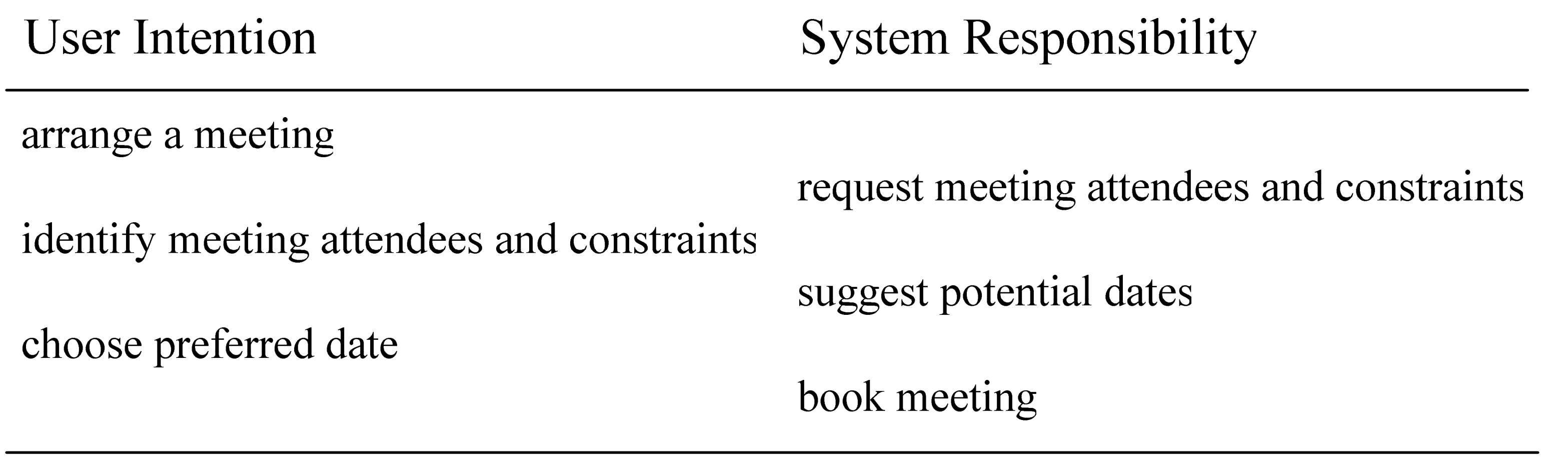
Difficult to capture concrete and specific activities while maintaining the generality required.

To Develop

Identify user roles, then examine these roles and identify the users’ goal or goals in using the system.

Task description and analysis, continued

- Essential use cases: the shared calendar example



User Intention	System Responsibility
arrange a meeting	
	request meeting attendees and constraints
identify meeting attendees and constraints	
	suggest potential dates
choose preferred date	
	book meeting

Source: *Interaction Design*, ch. 7

Task description and analysis, continued

Task Analysis

Description

Used to analyze the underlying rationale and purpose of what people are doing: what are they trying to achieve, why are they trying to achieve it, and how are they going about it, e.g. Hierarchical Task Analysis (HTA) & GOMS.

Advantages

Task analysis establishes a foundation of existing practices on which to build new requirements or to design new tasks.

Limitations

In the hands of inexperienced practitioners, too much level of detail may be entered into; for systems with diffuse objectives, time may be wasted by attempting to apply task analysis to intractable material. [2]

To Develop

Break a task down into subtasks and then into sub-subtasks and so on.

Task description and analysis, continued

- Task analysis: the shared calendar example (1 – text form)

0. In order to arrange a meeting
   1. compile a list of meeting attendees
   2. compile a lit of meeting constraints
   3. find a suitable date
      3.1 identify potential dates from departmental calendar
      3.2 identify potential dates from each individual's calendar
      3.3 compare potential dates
      3.4 choose one preferred date
   4. enter meeting into calendars
   5. inform meeting participants of calendar entry

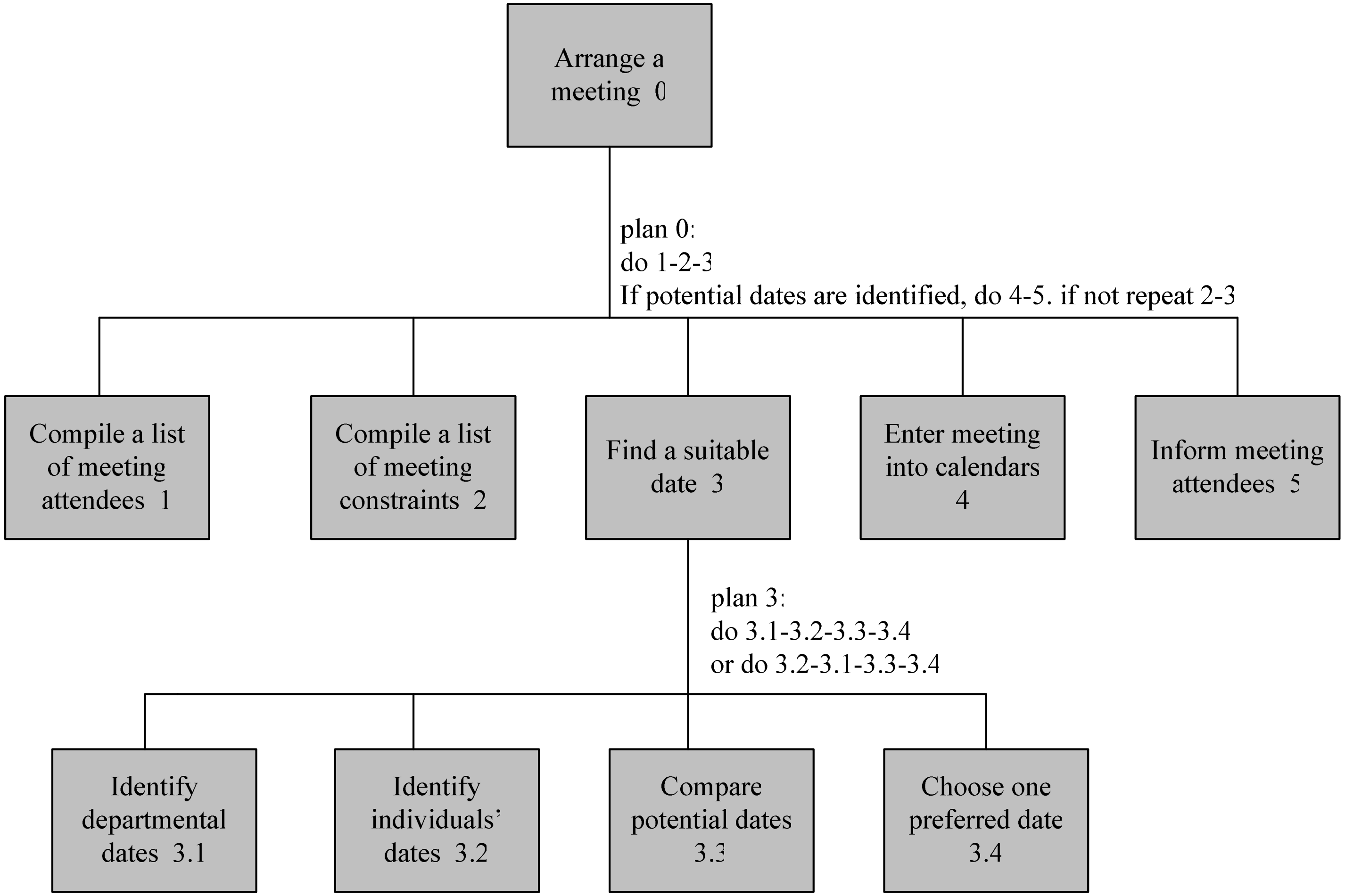
plan 0: do 1-2-3. If potential dates are identified, do 4-5. If no potential dates can be identified, repeat 2-3.

plan 3: do 3.1-3.2-3.3-3.4 or do 3.2-3.1-3.3-3.4

Source: *Interaction Design*, ch. 7

Task description and analysis, continued

- Task analysis: the shared calendar example (2 – diagram form)


Arrange a meeting 0
plan 0:
do 1-2-3
If potential dates are identified, do 4-5. if not repeat 2-3
Compile a list of meeting attendees 1
Compile a list of meeting constraints 2
Find a suitable date 3
Enter meeting into calendars 4
Inform meeting attendees 5
plan 3:
do 3.1-3.2-3.3-3.4
or do 3.2-3.1-3.3-3.4
Identify departmental dates 3.1
Identify individuals' dates 3.2
Compare potential dates 3.3
Choose one preferred date 3.4

Source: *Interaction Design*, ch. 7

Task description and analysis, continued

- Developers-centered descriptions: more formal, more specialized [4]
  - Entity-Relationship diagrams
  - Class diagrams
  - Ontologies
  - Goals
  - Finite State Machines

Summary

- "Getting the requirements right is crucial to the success of the interactive product."
- There are different types of requirements:
  - Functional, data, environmental, user and usability.
  - Every system will have requirements under each of these headings.
- Most commonly used data-gathering techniques for establishing requirements include:
  - Questionnaires, interviews, workshops or focus groups, naturalistic observation, and studying documentation.
- Describing user tasks such as scenarios, use cases and essential use cases can help to articulate existing user work practices.
  - They also help to express envisioned use for new devices.
- Task analysis techniques help to investigate an existing situation, i.e. existing systems and current practices.

Additional References

1. Sommerville, Ian; *Software Engineering, Sixth Edition;* Addison-Wesley, Boston, MA (2000).

2. *Scenario Building*, http://www.ucc.ie/hfrg/projects/respect/urmethods/scenario.htm

3. Schneiderman, Ben and Plaisant, Catherine; *Designing the User Interface, Fourth Edition;* Addison-Wesley, Boston, MA (2005).

4. van Vliet, Hans; *Software Engineering: Principles and Practice, First Edition*; John Wiley & Sons, Ltd (2000).

BREN:ICS

INFORMATION AND COMPUTER SCIENCES