User-Centered Design and Development

Instructor: **Franz J. Kurfess**

Computer Science Dept.

Cal Poly San Luis Obispo

Copyright Notice

- These slides are a revised version of the originals provided with the book "Interaction Design" by Jennifer Preece, Yvonne Rogers, and Helen Sharp, Wiley, 2002.

- I added some material, made some minor modifications, and created a custom show to select a subset.
  - Slides added or modified by me are marked with my initials (FJK), unless I forgot it …

484-W09 Quarter

- The slides I use in class are in the Custom Show "484-W09". It is a subset of the whole collection in this file.

- Week 4 contains slides from Chapters 6 and 7 of the textbook.

- The original slides are a bit of a mess, and I cleaned up various issues
  - outline view didn't show body text
  - quite a bit of "manual" formatting (bulleted/numbered lists)

Chapter 7

Needs and Requirements

Chapter Overview

- Needs
- Requirements
- Data Gathering
- Data Interpretation and Analysis
- Task Description
- Task Analysis

Motivation

- for user-centered design, it is critical to know the needs of the user
- these needs are often described more formally through requirements
- since users are not always able to express their needs directly, data gathering techniques are used to collect information
- the collected information needs to be interpreted and analyzed

Objectives

- gain experience identifying user needs and transforming them into requirements

- be familiar with various techniques for data gathering, interpretation, and analysis

- learn to utilize methods for developing task descriptions

Identifying needs and establishing requirements

Overview

- The importance of requirements
- Different types of requirements
- Data gathering
- Task descriptions:
  - Scenarios
  - Use Cases
  - Essential use cases
- Task analysis: HTA

What, how and why?

- What:
  - Two aims:
  1. Understand as much as possible about users, task, context
  2. Produce a stable set of requirements

- How:
  - Data gathering activities
  - Data analysis activities
  - Expression as 'requirements'
  - All of this is iterative

What, how and why?

- Why:
  - Requirements definition: the stage where failure occurs most commonly
  - Getting requirements right is <u>crucial</u>

Establishing requirements

-
  - What do users want? What do users 'need'?
    - Requirements need clarification, refinement, completion, re-scoping
    - Input: requirements document (maybe)
    - Output: stable requirements
  - Why 'establish'?
    - Requirements arise from understanding users' needs
    - Requirements can be justified & related to data

Different kinds of requirements

- Functional:
  - What the system should do
  - Historically the main focus of requirements activities
- Non-functional
  - memory size, response time...
- Data:
  - What kinds of data need to be stored?
  - How will they be stored (e.g. database)?

Different kinds of requirements

- Environment or context of use:
  - physical: dusty? noisy? vibration? light? heat? humidity? …. (e.g. OMS insects, ATM)
  - social: sharing of files, of displays, in paper, across great distances, work individually, privacy for clients
  - organisational: hierarchy, IT department's attitude and remit, user support, communications structure and infrastructure, availability of training

Different kinds of requirements

- Users: Who are they?
  - Characteristics: ability, background, attitude to computers
  - System use: novice, expert, casual, frequent
  - Novice: step-by-step (prompted), constrained, clear information
  - Expert: flexibility, access/power
  - Frequent: short cuts
  - Casual/infrequent: clear instructions, e.g. menu paths

Different kinds of requirements

- Usability:
  - learnability
  - throughput
  - flexibility
  - attitude

Note that *user* requirements and *usability* requirements refer to different things

Exercise: Kinds of requirements

What factors (environmental, user, usability) would affect the following systems?

- Self-service filling and payment system for a petrol (gas) station
- On-board ship data analysis system for geologists searching for oil
- Fashion clothes website

Activity: Requirements

- Select a task, and try to identify different kinds of requirements
  - functional/non-functional
  - data
  - context
  - environment
  - user characteristics
  - usability

Activity: Task Suggestions

- video downloading application
- voice interaction with household devices
- touch-screen ordering system for cafeteria ordering
- navigation system for motorcycles

Data Gathering Techniques

- Questionnaires
- Interviews
- Workshops and Focus Groups
- Naturalistic Observation
- Studying Documentation

Questionnaires

- A series of questions designed to elicit specific information
  - questions may require different kinds of answers:
    - simple YES/NO
    - choice of pre-supplied answers; comment
- Often used in conjunction with other techniques
- Can give quantitative or qualitative data
- Good for answering specific questions from a large, dispersed group of people

Interviews

- Forum for talking to people
- Structured, unstructured or semi-structured
- Props can be used in interviews
  - e.g. sample scenarios of use, prototypes
- Good for exploring issues
- But are time consuming and may be infeasible to visit everyone

Workshops and Focus Groups

- Group interviews
- Good at gaining a consensus view and/or highlighting areas of conflict

Observation

- Spend time with stakeholders
  - in their day-to-day tasks, observing work as it happens
- Gain insights into stakeholders' tasks
- Good for understanding the nature and context of the tasks
- Requires time and commitment from a member of the design team
- Can result in a huge amount of data
- Ethnography is one form

Studying Documentation

- Procedures and rules are often written down in manuals
- Good source of data about
  - steps involved in an activity
  - regulations governing a task
- Not to be used in isolation
- Good for understanding legislation, and getting background information
- No stakeholder time
  - a limiting factor on the other techniques

Choosing between techniques

- Data gathering techniques differ in two ways:
  - Amount of time, level of detail and   risk associated with the findings
  - Knowledge the analyst requires
- Also affected by the kind of task to be studied:
  - Sequential steps or overlapping series of subtasks?
  - High or low, complex or simple information?
  - Task for a layman or a skilled practitioner?

Activity: Data Gathering Techniques

- for each of the following data gathering techniques, try to find situations or tasks where it is particularly appropriate:
  - questionnaires
  - interviews
  - workshops and focus groups
  - naturalistic observation
  - studying documentation

Data Gathering Problems (1)

- Identifying and involving stakeholders
  - users, managers, developers, customer reps?, union reps?, shareholders?

- Involving stakeholders
  - workshops, interviews, workplace studies, co-opt stakeholders onto the development team

- 'Real' users, not managers
  - traditionally a problem in software engineering, but better now

Data Gathering Problems (2)

- Requirements management
  - version control, ownership
- Communication between parties
  - within development team
  - with customer/user
  - between users... different parts of an organisation use different terminology
- Domain knowledge distributed and implicit:
  - difficult to dig up and understand
  - knowledge articulation: how do you walk?
- Availability of key people

Data Gathering Problems (3)

- Political problems within the organisation

- Dominance of certain stakeholders

- Economic and business environment changes

- Balancing functional and usability demands

Some basic guidelines (1)

- 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

Some basic guidelines (2)

- Support the process with props
  - prototypes
  - task descriptions
- Run a pilot session
- You will need to compromise
  - on the data you collect
  - the analysis to be done,
  - but before you can make sensible compromises, you need to know what you'd really like
- Consider carefully how to record the data

Data interpretation and analysis

- Start soon after the data gathering session
- Initial interpretation before deeper analysis
- Different approaches emphasize different elements
  - class diagrams for object-oriented systems
  - entity-relationship diagrams for data intensive systems

Task descriptions

- Scenarios
  - an informal narrative story, simple, 'natural', personal, not generalizable
- Use cases
  - assume interaction with a system
  - assume detailed understanding of the interaction
- Essential use cases
  - abstract away from the details
  - does not have the same assumptions as use cases

Scenario for shared calendar

"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 departmental 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 people's 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."

Use case for shared calendar

- The user chooses the option to arrange a meeting.
- The system prompts user for the names of attendees.
- The user types in a list of names.
- The system checks that the list is valid.
- The system prompts the user for meeting constraints.
- The user types in meeting constraints.
- The system searches the calendars for a date that satisfies the constraints.
- 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 of them appointment

Alternative courses for shared calendar

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.

Activity: Task Descriptions

- use the following methods to develop task descriptions for the topic you chose in Assignment 2
  - scenarios
  - use cases
  - essential use cases
- in a team, you can split up and use the methods in parallel
  - ideally, there should be a significant overlap
  - probably also differences
    - different methods used
    - different people involved

Example use case diagram for shared calendar
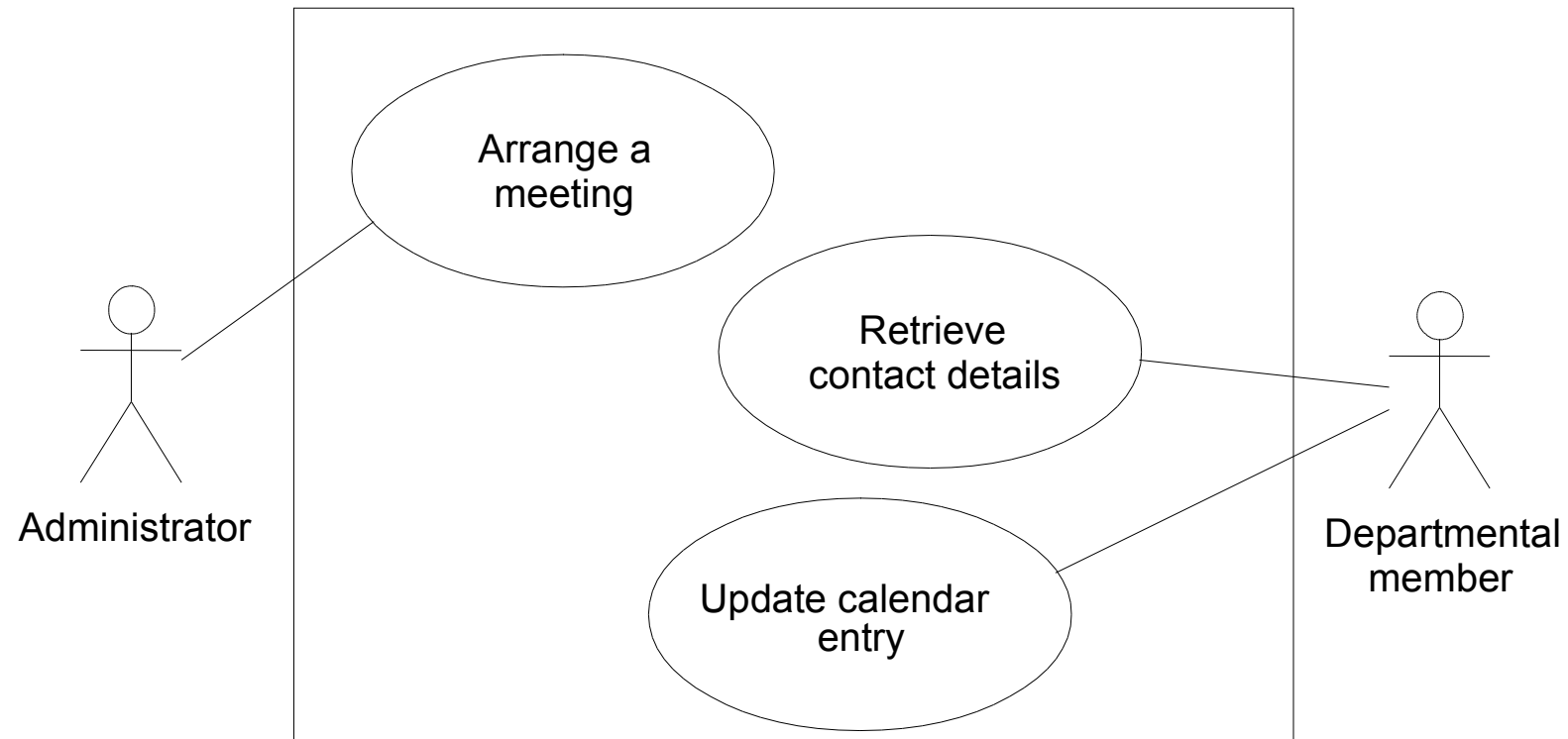
Example essential use case for shared calendar

arrangeMeeting

USER INTENTION	SYSTEM RESPONSIBILITY
arrange a meeting	
	request meeting attendees & constraints
identify meeting attendees & constraints	
	search calendars for suitable dates
	suggest potential dates
choose preferred date	
	book meeting

Task analysis

- Task descriptions are often used to envision new systems or devices

- Task analysis is used mainly to investigate an existing situation

- It is important not to focus on superficial activities
  What are people trying to achieve?
  Why are they trying to achieve it?
  How are they going about it?

- Many techniques, the most popular is Hierarchical Task Analysis (HTA)

Hierarchical Task Analysis

- Involves breaking a task down
  - into subtasks, then sub-sub-tasks and so on.
  - These are grouped as plans which specify how the tasks might be performed in practice
- focuses on physical and observable actions
  - includes looking at actions not related to software or an interaction device
- Start with a user goal
  - which is examined and the main tasks for achieving it are identified

Example Hierarchical Task Analysis

0.   In order to borrow a book from the library
    1.   go to the library
    2.   find the required book
        2.1 access library catalogue
        2.2 access the search screen
        2.3 enter search criteria
        2.4 identify required book
        2.5 note location
    3.   go to correct shelf and retrieve book
    4.   take book to checkout counter

Example Hierarchical Task Analysis (plans)

plan 0: do 1-3-4. If book isn't on the shelf expected, do 2-3-4.

plan 2: do 2.1-2.4-2.5. If book not identified do 2.2-2.3-2.4.

Example Hierarchical Task Analysis (graphical)

```
                    ┌─────────────────┐
                    │ Borrow a        │
                    │ book from the   │
                    │ library      0  │
                    └─────────────────┘
```

plan 0:
do 1-3-4.
If book isn't on the shelf expected, do 2-3-4.

```
┌─────────────┐  ┌─────────────┐  ┌─────────────┐  ┌─────────────┐
│ go to the   │  │ find required│ │ retrieve book│ │ take book to│
│ library     │  │ book        │  │ from shelf  │  │ counter     │
│          1  │  │          2  │  │          3  │  │          4  │
└─────────────┘  └─────────────┘  └─────────────┘  └─────────────┘
```

plan 2:
do 2.1-2.4-2.5.
If book not identified from information available, do 2.2-2.3-2.4-2.5

```
┌──────────┐ ┌──────────┐ ┌──────────┐ ┌──────────┐ ┌──────────┐
│ access   │ │ access   │ │ enter    │ │ identify │ │ note     │
│ catalog  │ │ search   │ │ search   │ │ required │ │ location │
│      2.1 │ │ screen   │ │ criteria │ │ book     │ │      2.5 │
│          │ │     2.2  │ │     2.3  │ │     2.4  │ │          │
└──────────┘ └──────────┘ └──────────┘ └──────────┘ └──────────┘
```
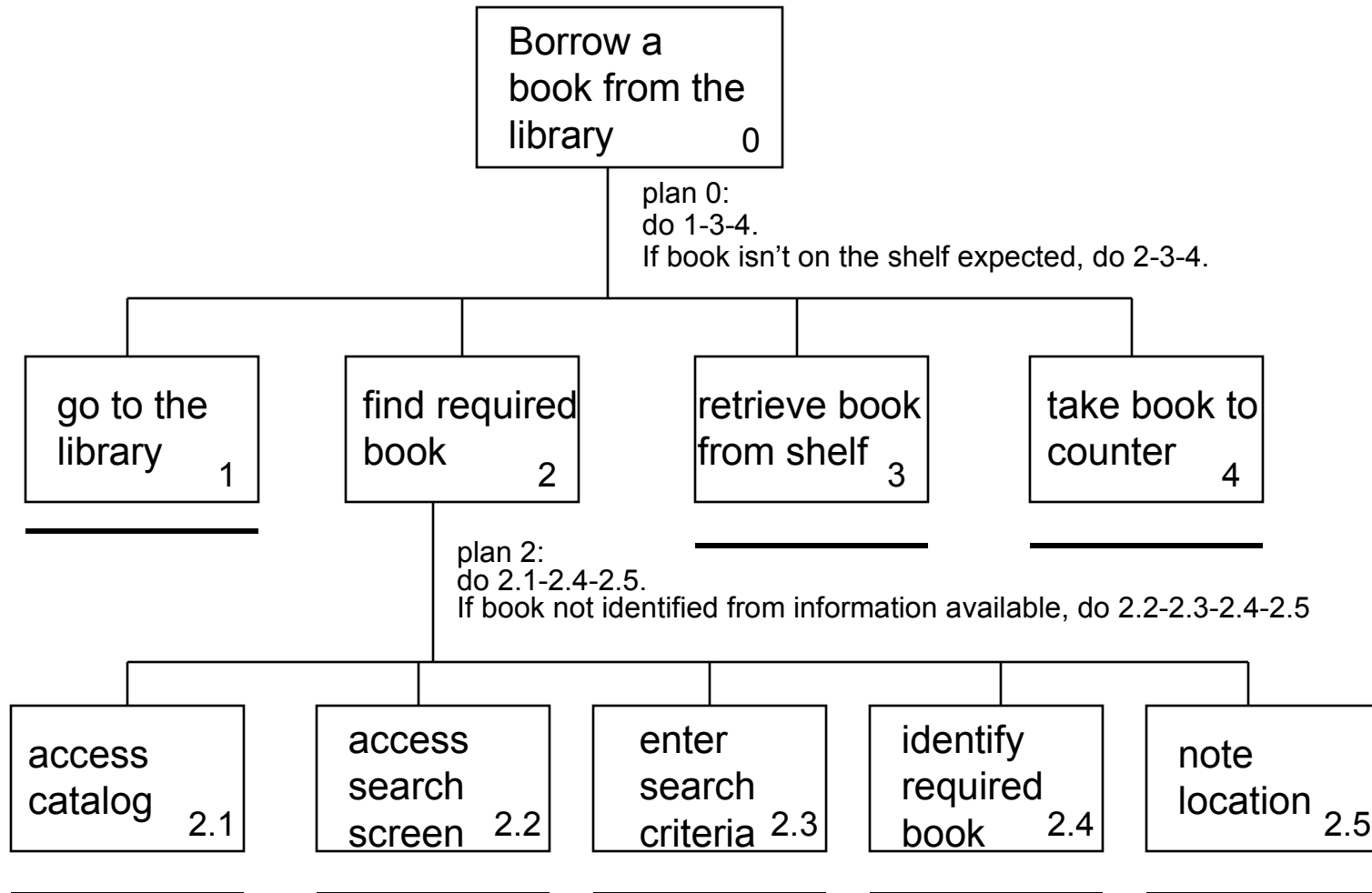
Activity: Task Analysis

- apply hierarchical task analysis (HTA) to the task you selected for Assignment 2

Summary

- Getting requirements right is crucial
- There are different kinds of requirements
  - each is significant for interaction design
- The most commonly-used techniques for data gathering are
  - questionnaires, interviews, focus groups and workshops, naturalistic observation, studying documentation
- Scenarios, use cases and essential use cases can be used to articulate existing and envisioned work practices.
- Task analysis techniques such as HTA help to investigate existing systems and practices