Outline (continued from Mon.)

1. Why, What and How of Requirements Gathering (last class)
2. Methods for Requirements Analysis and Requirements Gathering (last class)
3. User Characteristics and Usage Environment (last class)
4. Making Sense of All the Data
5. Documenting the Results
Making Sense of all the Data
Goal - Organize Information

• Organizing the information blends in with step 5, documenting the results
• Group process - compare notes
• Several tools can be used to facilitate
Tools for Making Sense

• Card Sorting - to create Affinity Diagrams

• Card sorting also useful for web site organization
  – Do it with multiple users
Affinity Diagram

• Write down each quote/observation on a slip of paper or post-it
• Put up on board
• Coalesce items that have affinity
  – If the users are saying similar things about an issue
• Give names to different groups (colors too)
• Continue grouping subgroups
• A hierarchy will be formed
Affinity Diagram - from Card Sorting

• Cards are notes from interviews, focus groups, etc.

Figure 9.4 A portion of the affinity diagram for the WordPerfect fieldwork studies.

From Interaction Design, Preece Rogers and Sharp
More Tools for Making Sense

• Flow charts and work flow diagrams
• Software tools for
  – Card sorting
  – Flow charts
  – Task analysis diagramming
Documenting the Results
Results of Requirements Gathering & Task Analysis

1. Real-world constraints
2. Essential Use Cases (Scenarios)
3. Usability Requirements
4. User characteristics + Personas
5. Hierarchical Task Decomposition/Task Model
   - A (typically) diagrammatic representation of a task - subtask decomposition
   - Includes sequencing information
6. Possibly workflow diagrams, flow charts
7. ER Model - Objects, properties of objects, operations on objects, relations between objects
2: Essential Use Cases (Scenarios)

• Description of important or frequent user interactions
• Used to evaluate / walkthrough various design alternatives
• Three elements
  – Name
  – User intention
  – System responsibility
• Do NOT make assumptions about the UI design
## Ex: Essential Use Case

### Arrange-Meeting

<table>
<thead>
<tr>
<th>USER INTENTION</th>
<th>SYSTEM RESPONSIBILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrange a meeting</td>
<td>Request meeting attendees and constraints</td>
</tr>
<tr>
<td>Identify meeting attendees and constraints</td>
<td>Suggest potential dates</td>
</tr>
<tr>
<td>Choose preferred date</td>
<td>Book meeting</td>
</tr>
</tbody>
</table>
Ex: Essential Use Case

<table>
<thead>
<tr>
<th>User’s purpose</th>
<th>System responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify self.</td>
<td>Validate user’s identity.</td>
</tr>
<tr>
<td></td>
<td>Display currencies available.</td>
</tr>
<tr>
<td>Select currency required.</td>
<td>Display exchange rate.</td>
</tr>
<tr>
<td>Enter amount of foreign currency required.</td>
<td>Calculate exchange (e.g. £ to $).</td>
</tr>
<tr>
<td>Confirm amount.</td>
<td>Request initiation of payment.</td>
</tr>
<tr>
<td></td>
<td>Obtain authorisation for amount.</td>
</tr>
<tr>
<td></td>
<td>Give money.</td>
</tr>
<tr>
<td>Take money and go.</td>
<td></td>
</tr>
</tbody>
</table>

From *User Interface Design and Evaluation*, The Open University
3: Usability Requirements

• Time to complete key tasks - min, max
• Time to become proficient - do given set of tasks in given time
• Subjective satisfaction

• MUST have this at start
  – Refine for later testing
## Ex: Usability Requirements

<table>
<thead>
<tr>
<th>Usability Attribute</th>
<th>Current Level</th>
<th>Worst Value</th>
<th>Target Value</th>
<th>Best Value</th>
<th>Observed Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Use: Time to edit a given document during first use of system</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Learnability: Time to edit a given document after one hour of use</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time to create outline</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subjective first impression</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4: Characterizing Users

<table>
<thead>
<tr>
<th>User characteristic</th>
<th>ATM customer characteristics, by group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Teens/Young adults</td>
</tr>
<tr>
<td>Age</td>
<td>12 to 25.</td>
</tr>
<tr>
<td>Sex</td>
<td>Both male and female.</td>
</tr>
<tr>
<td>Physical limitations</td>
<td>May be fully able-bodied, or may have some physical limitations in relation to, for example, hearing or sight. Will be of varying heights.</td>
</tr>
<tr>
<td>Educational background</td>
<td>May have minimal or no educational qualifications.</td>
</tr>
<tr>
<td>Computer/IT use.</td>
<td>Probably have some prior experience of computer or IT use.</td>
</tr>
<tr>
<td>Motivation</td>
<td>Probably very motivated to use the ATM, especially in relation to their banking habits.</td>
</tr>
<tr>
<td>Attitude</td>
<td>Attitudes to use may vary, depending on the services the automated teller offers and the reliability of the technology itself.</td>
</tr>
</tbody>
</table>

From *User Interface Design and Evaluation*, The Open University.
User Characteristics + Persona

• Description of user and what user wishes to do
• Be specific/detailed, even give names and picture
• Three personas for ATM usage follow
  – Adapted from *User Interface Design and Evaluation*, The Open University
• Developed by Cooper (1999)
Felix (representing teenage ATM users)

Felix is 13 and gets pocket money each week. He spends it with his friends, so doesn’t make regular deposits. He does receive gifts for his birthday, Christmas, etc. and saves that money for special purchases, such as a computer games console or trendy clothes. He has an ATM card allowing him to make withdrawals when needed for his purchases.
Sandra (representing young adults thru middle age)

Sandra is 30, is married to Jason, has two children Todd (6) and Carly (18 months). They live in a subdivision that is about three miles from the town center, where the bank and stores are located. Jason uses the car for work, and works long hours, leaving at 6:45 am and returning at 8:00 pm. Sandra does not drive, so has to use public transportation. She tries to run errands and shop while Todd is in school, so she does only has to take Carly to town with her. She typically needs to make two trips to town each week to get everything done. She uses a stroller with Carly, and the bank is one flight up via escalator, so she prefers to use the ATM outside the first floor, even though there is no canopy to protect customers from bad weather.
Grandpa Marvin (representing middle age to senior citizens)

Marvin is 68 years old, and his social security is deposited into his bank account at the start of each month. He goes to the bank every week, withdrawing enough cash for the week - for miscellaneous expenditure. Regular bills are paid by check. He stands in line for a live teller, as he prefers the social interaction to using an ATM, even though his new artificial hip makes standing in line uncomfortable. He does not have an ATM card.
5: Hierarchical Task Decomposition/Task Model

• Goals – what the user wants to achieve
• Tasks – do these to achieve the goals
  – Sequential dependencies
    • Ex: Create new document before entering text
  – Multiple occurrences of tasks
  – Subtasks – lower-level tasks
• The lowest-level subtasks get mapped onto one or several UI commands
  – ie, move done by a copy followed by a paste
Example Task Clusters

- Fixed sequence
- Optional tasks
- Waiting events
- Cycles
- Time-sharing
- Discretionary
Ex: “Borrow Book” Task Model

- Sequences added as annotations
- Can also show hierarchy as indented text

Figure 7.12 A graphical representation of the task analysis for borrowing a book.

From Interaction Design, Preece Rogers and Sharp
Ex: “Write a Letter” Task Model

This process can go arbitrarily deep
Stop when not useful to go further - but not before!
Is this model done?
Alternative Sequences

• How X writes a letter
  – Get an envelope, paper, pencil, stamp
  – Write letter
  – Address the envelope
  – Apply stamp
  – Put letter in envelope
  – Seal envelope

• How Y writes a letter
  – Get an envelope, paper, pencil, stamp
  – Address the envelope
  – Write the letter
  – Put letter in envelope
  – Apply stamp
  – Seal envelope

• How does Z do it?
Ex: “Schedule Meeting” Task Model

Figure 7.13 A graphical representation of the meeting HTA.

From Interaction Design, Preece Rogers and Sharp
Task Model as an Outline

- Lists, outlines, matrices
  - Use expanding/collapsing outline tool
  - Add detail progressively
  - Know in advance how much detail is enough
  - Can add linked outlines for specific subtasks
  - Good for sequential tasks
  - Does not support parallel tasks well
  - Does not support branching well
Ex: Task Model as an Outline

Using a lawnmower to cut grass

Step 1. Examine lawn
  - Make sure grass is dry
  - Look for objects laying in the grass

Step 2. Inspect lawnmower
  - Check components for tightness
    - Check that grass bag handle is securely fastened to the grass bag support
    - Make sure grass bag connector is securely fastened to bag adaptor
    - Make sure that deck cover is in place
    - Check for any loose parts (such as oil caps)
    - Check to make sure blade is attached securely
  - Check engine oil level
    - Remove oil fill cap and dipstick
    - Wipe dipstick
    - Replace dipstick completely in lawnmower
    - Remove dipstick
    - Check that oil is past the level line on dipstick

...
Task Model as a Narrative

• Describe tasks in sentences
• Often expanded version of list or outline
• More effective for communicating general idea of task
• Not effective for
  – details
  – branching tasks
  – parallel tasks
• GREAT as introduction to diagrams or outlines
6: Workflow

• Documents going from one person/organization to another

• Multiple participants in an activity
Ex: “Document Flow” Workflow

Create Travel Request (Traveler) → Ensure Funds Available (Accounting) → Approval (Dean) → Notification of Approval (Dean)

No Funds

Notification of Approval (Dean) → Make Trip (Traveler) → Complete Expense Report (Traveler) → Approval (Accounting) → Etc
Workflow Example - Multiple Participants

Figure 9.5 An example workflow model.

From *Interaction Design*, Preece Rogers and Sharp
Flow Charts

• Flow Chart of Task Steps
  – Complete, can become complex
    • Sequential flow, branching, parallel tasks.
  – Includes actions, decisions, logic, by all elements of the system
  – Mature, well-known, good tools for doing it
Flow Chart Example

1. Start
2. Continue?
   - Y: Document
   - N: Display
3. Manual Operation
4. Input
5. End
7: Object (E-R) Models

• Network / Entity-Relationship Diagrams
  – Objects/people with links to related objects
    • Stress relationship between objects and actions
  – Links described functionally and in terms of strength
    • Task: Develop design for final project
      – objects - pens, paper, drawing tools, etc.
      – actors - Mary, Bob, Sally
      – composite objects - the “team”

• About relations, not procedures
  – Complements Hierarchical Task Analysis & flow charts
Object Model: Simple Drawing System

• Objects
  – page, line, point

• Relations
  – page contains zero or more lines and points
  – Lines defined by two points

• Actions on objects
  – Page: clear
  – Points: create, delete, move
  – Lines: create, delete, move

• Etc
Object Model: Line Text Editor

• Objects
  – Files, lines, characters

• Relations
  – File is sequence of lines
  – Line is sequence of characters

• Actions on objects
  – Files: create, delete, rename
  – Lines: create, delete, move, copy
  – Characters: insert, delete, move, copy
Object Model

• What would be the model for a string editor rather than a line editor?
• How about for a WYSIWYG editor like Microsoft Word?
• Similar to data model, but includes operations.
• Operations are not necessarily the UI commands
Object Model - Other Typical Elements

• Relations
  – X is a set of Y
  – X is a sequence of Y
  – X is made up of (A, B, C)
  – X is geometrically aligned with Y

• Actions on relations
  – Remove X from set or sequence
  – Insert Y into set or sequence

• Actions on attributes
  – Set, modify, inquire
Summary

• Determine what data you need (last class)
• Gather it using various appropriate methods and techniques (last class)
• Represent the tasks and subtasks, plus other related information (today)
• Use this data as basis for design (part 2)

• Note: Be efficient!