INTRODUCTION TO EXAMPLE SYSTEM

• Calendar Management System
  * Simple automated version of a paper calendar
  * To illustrate user interaction development process activities
  * Goal is to learn the development process, not to produce a marketable calendar product
INTRODUCTION TO SYSTEMS ANALYSIS

• Systems analysis
  * Business process modeling
  * Needs analysis
  * User analysis
  * Task analysis
  * Functional analysis
  * Task / function allocation
  * Usability goals
  * Requirements
  * Iteration
EXAMPLE:
SYSTEMS ANALYSIS

• Goal:
  * To make a fast tour through the process of determining basic user and system requirements

• Activities:
  * A mixture of needs analysis, user modeling, task analysis, functional analysis, task/function allocation, usability goals, and requirements definition
BUSINESS PROCESS MODELING

• Understand application domain

• Must begin by understanding the client's business process
  * This is good starting point for interaction design
  * Important for non-UI software development, too (functionality)
  * Goal is to capture
    - What gets done to run the business
    - How it gets done
    - How it relates to other things that get done
  * How to capture it
    - Look for both computer-supported and non-computer tasks
    - Gather and use work artifacts (e.g., paper work, tickets, slips)
    - Find work flow, task flow, data/document flow
    - Flow charts are good (e.g., tasks as boxes)
EXAMPLE: 
NEEDS ANALYSIS

• *Goal of system*: Manage appointments

• Assumption: Some boundaries set by management, marketing, customer, etc. (e.g., hardware); determination made that product is novel, market not yet saturated

• Features

  * *Appointment* means information on:
    Date  
    Time  
    Place  
    Appointment description

  * *Manage* means:
    Add new appointment  
    Delete existing appointment  
    Modify existing appointment

  * Plus, need ability to view/display appointments

• Follows from user analysis, not just developers’ ideas
EXAMPLE:
NEEDS ANALYSIS

• Some time later, someone thinks of “alarm” idea
  * Do we want to actively inform of appointments?
  * Decision: Yes, very useful
  * Iterate and revise needs

• New feature: Active reminder
USER ANALYSIS

- What are characteristics of users of this system?

  * General characteristics
    - Busy people
    - Keep schedule for self and others
    - Professional and personal use
    - Calendar is very small part of job
    - Need ‘transparent’ tool
    - High general skill level, literate

  * Domain skills
    - Know how to use calendar

  * Computer skills
    - Broad range
    - At least some typing skills
    - Familiar with GUI/mouse
EXAMPLE:
USER ANALYSIS

• Conclusions
  * Keep it simple
  * Usability as important as functionality (or more)
  * Try to get functionality greater than paper calendar
  * Try to get usability greater than paper calendar
  * Minimize typing
  * Users must learn it quickly

• Caution: Difficult for users to tell developers what they want or need
  * Important to observe users in their typical work environment

Copyright © H. Rex Hartson and Deborah Hix. All rights reserved.
SA 8
USER CLASS CHARACTERIZATION

• Toward more "standard" user class definitions

  * User group — user class, set of users with similar characteristics and needs

  * Group based on many different dimensions (e.g., job description, location, level of responsibility, computer literacy, application domain knowledge)

• Developers do not know all this information; must interact with representative user(s) to discover it
USER CLASS CHARACTERIZATION

• Answer following questions for each class:
  * What is knowledge of computer technology?
  * What is knowledge of application domain?
  * How complex is application content?
  * Are users discretionary or captive?
  * With whom do users interact?
  * What are training needs?
USER CLASS CHARACTERIZATION

• Answer following questions for each class (continued):
  * What is user culture?
  * How receptive/resistant to product are users?

• Simply answering these questions about initial set of user classes will probably uncover more user classes!

• Much of rest of systems analysis is done for each user class (e.g., task analysis, usability goals, usability specifications)
## USER CLASS CHARACTERIZATION MATRIX

<table>
<thead>
<tr>
<th></th>
<th>User Segment A</th>
<th>User Segment B</th>
<th>User Segment C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer knowledge</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Domain knowledge</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complexity of content</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discretionary or captive?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>With whom do they interact?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usage frequency?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>What info is exchanged (human-to-human)?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Culture?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receptive/resistant?</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
EXAMPLE:
TASK ANALYSIS

• What tasks will users perform with this system?
  * Tasks are performed by user (e.g., view)
  * Initial list of major sub-tasks
    - Add new appointment
    - View existing appointments
    - Modify existing appointments
    - Delete existing appointments
    - Set alarm
    - View calendar
  * This is task structure
  * Not timing, precedence, order of task performance, work flow, etc.
  * Only what user can do, not must do
EXAMPLE:
FUNCTIONAL ANALYSIS

• What functions will system perform?
  * Functions are performed by system (e.g., display)
  * System functions to support tasks
EXAMPLE:  
TASK ANALYSIS

• Task analysis iterated

* As think about viewing appointments, realize the need for different levels or scopes of view
  - For example, by month, by week, by day, by hour
  - Implication: Add “control view” task to list

* Also discovered need to search appointment database to retrieve by content
  - Implication: Add to needs, tasks, functions, requirements
  - Note: From here on, "requirements" means interaction design requirements

* Another example of iteration: Alarm feature will lead to user tasks (to set parameters)
  - Decision: For now, hard wire for 10 minutes before appointment; no user tasks
EXAMPLE:
TASK ANALYSIS

• What developers observe that users need, not what developers think that users need

• Task analysis is probably the most overlooked and shortchanged activity in the whole user interaction development process
EXAMPLE:
HIERARCHICAL TASK ANALYSIS

Example of possible hierarchical user task structure for Calendar Management System

- Hierarchical structure is accompanied by brief description of what each box means
USABILITY GOALS

• Project-specific usability goals identify objectives in terms of usability and design of user interaction
  * Reflect real use of product in real world
  * Determine what is important to organization and to users

• Usability evaluation design (later in semester) driven by usability goals
  * Will be the basis for setting usability specifications

• Determining usability goals in terms of
  * User classes and workgroups
  * User task context, special tasks
  * Walk-up-and-use learnability
  * High performance for expert users
  * User errors
USABILITY GOALS

• Errors
  * How important is it to avoid user errors?
  * What are consequences of errors?
  * How will errors be reduced or avoided?
  * Is performance in error recovery important?
INTRODUCTION TO TEAM EXERCISES

• **Goal:**

  * To learn how to perform activities in the user interaction development process, *not* to develop world's best Web application

• **Exercises**

  * Decide on an application system for your team's design project
  * Simple systems analysis
  * Scenarios and screen layout sketches
  * Usability specifications
  * Rapid prototyping
  * Formative evaluation

  => *Report to class on all of above*
TEAM EXERCISE: SYSTEMS ANALYSIS

• *Goal:*  
  * To develop as a team a partial rapid systems analysis

• *Assumptions:*  
  * Generic desktop/workstation platform
  * Can assume knowledge of GUIs and Web browsers in general
  * Don’t assume specialized knowledge of particular browser
  * Use our systems analysis as example for your application; i.e., keep it very simple

• *Activities:*  
  * As part of your needs analysis, begin by answering these questions:  
    - Why does this project need to use the Internet?  
    - Who is our audience/market?  
    - What are our users trying to accomplish?  
    - What is our competition doing?
- What are our business goals for the site?
TEAM EXERCISE:
SYSTEMS ANALYSIS

• Activities (continued):

* Perform simple user analysis to begin two or three user class definitions, using the LUCID matrix given

* Perform just a sampling of task analysis

  - Identify several key user tasks and subtasks

  - Draw at least the beginnings of a hierarchical task diagram

From what you know about the application, the users, and the tasks, come up with a few reasonable usability goals for your design
TEAM EXERCISE:
SYSTEMS ANALYSIS

• *Deliverables:*
  ∫ List of answers to needs analysis questions
  ∫ Filled out user class characteristics matrix (blank form on next page)
  ∫ List of key user tasks and subtasks (keep it simple)
  ∫ Hierarchical task diagram (abbreviated)
  ∫ List of usability goals

• *Completed by:*
### USER CLASS CHARACTERIZATION MATRIX

<table>
<thead>
<tr>
<th></th>
<th>User Segment A</th>
<th>User Segment B</th>
<th>User Segment C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer knowledge</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Domain knowledge</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complexity of content</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discretionary or captive?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>With whom do they interact?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usage frequency?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>What info is exchanged (human-to-human)?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Culture?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receptive/resistant?</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>