Lecture 4

Human-Centered Development

Three Approaches to UI Design

• Attitude of technology-centered development
  – Progress made by technological advances
  – The more bells & whistles the better

• Attitude of designer-centered development
  – Progress made by considering the intuitions of the designer
  – Imagining what the user will do and feel

• Attitude of human-centered development
  – Progress made by incorporating the users into the design process
  – Empirical studies integrated early into the design
Human-centered software development
(John Gould, IBM 1983)

- Definition
  - Early and Continued Focus on Users
    - Direct contact through interviews, observations, surveys, participative design in order to understand characteristics of users and their jobs
  - Integrated Design
    - All aspects of usability evolve in parallel; All aspects of usability under one focus or person
  - Early and Continued User Testing
    - Throughout development, intended users do real work with simulations and prototypes; their performance and reactions are measured qualitatively and quantitatively
  - Iterative Design
    - The system (functions, user interface, help system, training material, training approach) is modified based upon results of user testing; testing cycle is repeated

Iterative Model of Development

- Steps in method

1. Planning
   - Scope of project, investigate user population (document analysis, interviews, surveys, observation) & related systems
2. Requirements Analysis
   - Task analysis of existing system, scenario development
   - Requirements for usefulness (functionality) and usability
3. Design (Presentation & Interaction Design)
   - Specifications (yes?) for human-computer interaction (UI)
4. Implementation (Prototyping)
   - Storyboards, mock-ups, initial prototypes
5. Usability Evaluation
   - Evaluation without users: cognitive walkthrough, heuristic evaluation (guidelines), GOMS, Keystroke Level Model (KLM)
   - Evaluation with users (usability testing, interviews, questionnaires)
Step 1: Planning

- Scope of project
  - Time frame
  - Costs and other resources
  - Purpose
  - Context
- Gathering information about needed system
  - Prior and related systems
  - User studies
    - Who are the users?
    - What should the system do?
    - User needs
    - User constraints

User Studies

How do you get user information?

- Artifact analysis
- Interviews
- Observation
- Participation
- Survey/Questionnaire
Artifact Analysis

- Collect and examine the documents, objects and other resources that people use in their activities
- Try to understand the content of the information and the role it plays in activities

Interviews

- Advantages
  - Gathers opinion, Creates rapport
- Disadvantages
  - Must be well-planned
  - Bias: Information often filtered
- Types
  - Structured
    - Fixed set of questions with simple answers
  - Unstructured
    - More open questions

Observation

- Advantages
  - Not an opinion, but an objective record
  - Captures detail
- Disadvantages
  - Intrusive, Time consuming
- Types
  - Passive: “Hanging out”
    - Used in very early design when don’t have much information about user’s activities
  - Active
    - Provide users with problems to solve or tasks
Participation

- When the observer learns and participates in the work activities
- Overcomes the Hawthorne effect
  - Bias of being observed
- Obtain otherwise privileged information
- Creates first-hand domain knowledge

Survey/Questionnaire

- Purpose
  - Reaches lots of people
  - Perform statistical analysis on data
  - Avoids bias by anonymity
  - Consistent questions
- Design Issues
  - Must be carefully designed, do pilot
  - Must not be time-consuming & easy to reply
  - Must make sense
  - Must gather precise answers, not vague ones
  - Ask only questions which support the design
  - Sample vs. population

Survey: Good Example
Step 2: Requirements Analysis

- What is a requirement?
  - What the system will do, not how
  - Captures constraints as well
  - Types of requirements
    - User Functions: what the user can do
    - Usability: combines functions with usability measures
    - Other: hardware, software functions or constraints

Goals for requirements analysis

- **Ascertain the user’s needs**
  - Determine what tasks and subtasks must be carried out
  - Include tasks which are only performed occasionally. Common tasks are easy to identify.
  - Functionality must match need or else users will reject or underutilize the product

Getting User Requirements
Requirements

• Example: Rapid Transit Ticket Dispenser
  – Functional requirement: User must be able to purchase ticket
  – Usability requirement: User must be able to purchase simple ticket in under 2 minutes.
  – Functional requirement: Support for blind users.
  – Usability requirement: Blind user must be able to purchase simple ticket in under 4 minutes.
  – Functional requirement: User instructions should be in English and Spanish.
  – Usability requirement: User must be able to read instructions at 8th grade level.

Requirements as Tasks

• Example: Rapid Transit Ticket Dispenser
  – Function: Purchase ticket
    • Subfunction: Determine fare
      – Sub-subfunction: Give the destination
      – Sub-subfunction: Specify journey type, either one-way or round trip
      – Sub-subfunction: Receive quoted fare
    • Subfunction: Obtain ticket
      – Sub-subfunction: Pay the money
      – Sub-subfunction: Receive the ticket and any change due
  • NOTE: Hierarchical levels of abstraction
  • Task Analysis (Greenberg reading)

Step 3: Design

• Implements requirements (How system will work)
• Document proposed design as specification -- like a blueprint
  • Formalizes Design
    – Specification proposes the exact user interaction and presentation; leaves nothing to ad hoc decisions
  • Guides Implementation
    – Used to define the programming implementation
    » Widget types, Graphics and text, Error and help processing
  • Creates Communication
    – Represents the evolving design to the client and all members of the team
  • Evaluates design
    – completeness, correctness, consistency and performance times
Design Specification as a Task

- Example: Rapid Transit Ticket Dispenser
  - Function: Purchase ticket
    - Subfunction: Determine fare
      - Sub-subfunction: Give the destination
      - Sub-subfunction: Specify journey type, either one-way or round trip
      - Sub-subfunction: Receive quoted fare
    - Subfunction: Obtain ticket
      - Sub-subfunction: Pay the money
      - Sub-subfunction: Receive the ticket and any change due
  - Will be implemented as a network of dialog boxes on an ATM type machine (limited buttons, number keypad)

Design Specification Types

- Presentation specification
  - Initially low fidelity sketches progressing to screen shots

- Interaction specification
  - Describes the Tasks beginning with the core
  - Types
    - Narrative
    - Scenario
    -Storyboard
    - Network

A Picture Is Worth 1000 Words!
Interaction as Narrative

- **Definition**
  - Sentences describing the tasks: how the user accomplishes each functional requirement
- Reference within the text to a view of the presentation (screen)
- Can also represent tasks and actions as indented text
- Can include the context for the task and fictional users: scenario (Example in Greenberg reading)
- Can be used in user documentation for completed system

Interaction as Narrative

- **Example: Graphics editor**
  - To draw a Bezier curve, the user selects “Bezier” from the Draw menu (see Figure 1), and then chooses either “Point-to-point” or “Curve fit”. In point-to-point mode the user then clicks on positions in the drawing window. The system draws a Bezier curve fit to each point. (See Figure 2 for an example.)

![Figure 1](image)

Interaction as Storyboard

- **Definition**
  - a method developed by animators many years ago to design cartoons
- Storyboard shows the major “moments” in an animation as a sequence of pictures
- Storyboard can be used to show user interaction as a sequence of pictures of the screen
- May be annotated with comments
Storyboard Example: ATM Machine

- Type: Sequence of Screens
- Task: Fast Cash Money Withdrawal

ATM Screen 1a

The Exchange
Please insert your card
-face up-

ATM Screen 1b

The Exchange
Please insert your card
-face up-

Hello
Sarah Ann Douglas
Please enter your PIN
Press when done------>

Press Cancel if error made
ATM Screen 4
Select your transaction
Press cancel if error made
Fast Cash $40
Withdrawal
Balance
Next Selection

ATM Screen 8
Transaction being processed
Please wait

ATM Screen 5
Please remove your cash from tray
ATM Screen 6

Do you want another transaction?

Yes-------------->
No--------------->

ATM Screen 3

Please take card

Thank you!

ATM Screen 1

The Exchange
Please insert your card
-face up-
Beyond Storyboards to Interaction Networks

- Definition: Interaction network
  - Abstract representation of the interaction between the user input and the display
  - Augmented Transition Network (ATN)
    - Nodes
      - System response
    - Arcs
      - User action
      - Finite State machine

ATM Example

```
ATM Example

system response
user action
```

ATM Example

```
ATM Example

system response
user action
```
ATM Screen 2

Do you want more time?

Yes-------------------->
No------------------------>

ATM Example

Select Account for Balance inquiry

Press cancel if error made
Checking----------------->
Savings----------------->
Credit Card----------->
ATM Example

ATM Screen 9

Total
Available
$172.48

Press this key to continue

Example: Heartbeat
Heartbeat Main Menu

Example: Heartbeat

Heartbeat Set Preferences
Step 4: Implementation (Prototyping)

- Physical Storyboards
- Mock-ups
- Software prototypes

Step 5: Usability Evaluation

- Another whole lecture!