Review of Usability Process

- **Requirements**: understand users’ current activities well enough to reason about technology-based enhancements

- **Activity Design**: work from problems and opportunities of problem domain to envision new activities

- **Information Design**: identify methods for representing and arranging the objects and actions possible in a system in a way that facilitates perception and understanding

Interaction Design

Goal: *specify the mechanisms for accessing and manipulating task information*
Interaction Design

- Specifying the action sequences for planning and achieving one or more task goals
  - conveying what system goals are possible
  - support user plans for accomplishing them
  - design physical actions to execute
- Usability engineering of an interaction design
  - ensure users can predict how to pursue goals, and that doing so is a comfortable and pleasant experience
  - depends inherently on task, hence an important role of user interaction scenarios

Stages of Action in HCI

focus of information design

GULF OF EVALUATION

Last month’s budget...?

GULF OF EXECUTION

focus of interaction design
Interaction Design

1. Selecting a System Goal

2. Creating an Action Plan

3. Executing an Action

System Goals vs. User Goals

• System goals: software-oriented tasks
• Going from users’ task concept to system concept: the cognitive distance between two models

  – user’s mental model guides decisions & activity
  – user interface conveys designers’ model
  – the closer the match, the easier to identify and achieve a relevant goal
System Goals vs. User Goals

**User Goals**
- Check an account balance
- Make a payment of $100
- Check for new email

**System Goals**
- Login to online account summary
- Go to payment webpage, follow prompts and enter $100
- Start email program and click “Check for messages” icon

Suggesting Goals to the User

- Labeling: menu titles, folder names, application names, etc.
- Decreasing the distance via direct manipulation:
  - UI controls appear as *physical analogs* of real objects; their affordances suggest interaction goals
  - Examples: buttons, files and folders, tabs
  - Objects have visual representation that users interact with
  - Immediate and continuing interface feedback is provided
Opportunistic Behavior

- Visual or auditory UI elements sometimes lead to opportunistic selection of goals
  - interesting object or message intrudes on a task
  - user is paused, choosing among things to do; especially common among novice users
- Caution: can lead to usability problems
  - confusion
  - distraction

Command Languages

- Recall instead of recognition
- Commands refer to system objects and actions
  - cognitive distance determined by *words* and *phrases* a system understands
  - recall more demanding, but flexible & saves screen space
  - design issues: vocabulary size and structure, familiarity and ambiguity
  - also the *syntax* (grammar) of the command language
- Compromise: nested menus support hierarchy-path
  - *cued recall* of a large set of commands
  - e.g., *Format* > *Alignment* > *Center*
Interaction Design

1. Selecting a System Goal

2. Creating an Action Plan

3. Executing an Action

Action Planning

- Action plan: steps involved in accomplishing a system goal
- Some goals correspond to a conscious plan
  - e.g. complex tasks, novice user actions
- Goal is decomposed into subgoals, steps, etc.
- Experienced users think at a higher-level of abstraction
  - Example: action plan for changing to a document to be double-spaced?
Modeling Action Plans with GOMS

- **Goals**: unit task (in general, verb-noun)
  - e.g. delete word, change font, insert picture, spell check
- **Operators**: elementary user actions
  - observable actions with software
  - internal actions performed by user
- **Methods**: operator sequences to achieve a goal
- **Selection rules**: choose among different methods
  - Personal rules that decide which method to use
- Typically used to predict task efficiency

Key Stroke Model GOMS

- **Operators**
  - K: to press a key or button
  - P: to point with a mouse to a target on a display
  - H: to home hands on keyboard or other device
  - D: to draw a line segment on a grid
  - M: to mentally prepare to do an action or closely related series of primitive actions
  - R: to symbolize the system response time during which the user has to wait for the system
Example: GOMS model of Paragraph Delete

- GOAL: Delete Paragraph
- Method for goal: delete paragraph
  Step 1. Select Paragraph
  Step 2. Execute delete command
  Step 3. Return with goal accomplished.

Select Paragraph Alternatives

- Move cursor to top of paragraph (P)
- Press and hold button (K)
- Move cursor to end of paragraph (P)
- Release button (K)
- Position cursor within paragraph (P)
- Triple click (3K)

- Selection rule: If paragraph is long, then use method on right
Designing Learnable Action Plans

- Try to make the sequence of actions match how user thinks about the real world task
  - Use interaction widgets that have good affordances
  - Limit system task complexity
- Design a sequence, then analyze and refine
  - Limited storage capacity of short-term memory (7 +/- 2)
  - Look for ways to chunk long sequences of steps
  - Create subplan chunks that match task subgoals
  - Use intermediate feedback or physical actions to “create” boundaries in subplans

Example: Change Indentation

1. Specify text selection start
2. Specify text selection end
3. Select Format menu
4. Select Paragraph option
5. Set Special to FirstLine
6. Type value for FirstLine
7. Accept new settings
Guided Action Planning

- Using forms, dialogs, wizards, etc.
- Procedure (action plan) is implicit in the layout:
  - Numbering
  - Instructions
  - Tabs
  - Step-by-step actions

Multi-Threaded Interactions: Giving Control to the User

- Humans are good at, and expect support for, doing multiple things at once
  - Working in parallel or “stacking” then reinitiating a task
- Multiple windows, each with an application
  - Need to help users keep focused on priorities
  - Need to provide status information for windows
    - what they have done so far, what is possible, etc.
- Modes limit flexibility, only certain actions are allowed
  - e.g. preemptive dialog box that must be dismissed
  - when are modes good?
Interaction Design

1. Selecting a System Goal

2. Creating an Action Plan

3. Executing an Action

Action Execution

- Physical actions of dragging, clicking, typing, etc.
  - design goal is to minimize
  - movements agree with action plans

- Crucial role of *input device* - different devices have different affordances:
  - keyboard?
  - joystick?
  - steering wheel?
  - stylus?
  - wand?
  - pen and tablet?
What makes an input device “good”? 

• Each input device can correspond to multiple interaction techniques 
  – e.g. one-button mouse: single click, double click, triple click, click and drag, short click, long click, etc.

• Implementing techniques: 
  – Reuse the users’ exiting knowledge 
  – Ensure *articulatory directness*: 
    • Physical interactions with device correspond to task actions 
    • Examples: rotate by twisting, pick up and move by clicking and dragging 
  – Provide appropriate user interface feedback 
    • Examples: highlight selection, continuous update display based on movements, reflect keystrokes

Designing for Errors

• Key point: people always make errors 
  – “Read everything before doing anything!” 
  – *Why do people make errors?*

• Analyze physically actions 
  – Errors can reveal learning activities, less usable interactions, disastrous interactions

• Mistake – error due to a wrong intention 
  – Common in novice users

• Slip – right intention, but problem in execution 
  – Common in experts who “slip”
Supporting Error Management

- Disabling inappropriate commands
  - Graying out options
- Blocking inappropriate commands
  - Nothing happens or computer beeps
- Confirmation prompt
  - Do you really want to do this?
- Undo!
  - Reversibility, e.g. Back button
  - What should be allowed to be undone?
  - What level of granularity is used?
  - Others?

Optimizing Execution Sequences

- Execution of a task should be efficient!
- Feedback and good defaults are essential
  - especially in long, costly, or tedious transactions
- Consider implications of longterm use
  - focus on actions for frequent choices, fast-paths
  - BUT, be careful to notice inconsistencies with less frequent tasks
- Customization: users define their own sequences
  - e.g., mapping commands to key combinations
  - can be critical when supporting users with special needs
- Consider # of interactions and GOMS analysis