SYST 542
Decision Support Systems Engineering

Instructor: Kathryn Blackmond Laskey

Fall Semester, 2006

Unit 5: DSS Elements: The Dialog Subsystem

Outline

- Functions of dialog subsystem
- Some interaction guidelines for usable systems
- Design for usability

Why the Dialog Subsystem is Important

- The user is ultimately responsible for making the decision
- A DSS will not be used if its input requirements are overly burdensome
- A DSS will be useless if the user cannot process its output and integrate with his/her decision making process

Functions of Dialog Subsystem

1. **Data entry**
2. **Information display**
   Organize information to facilitate comprehension and effective use
3. **Sequence control**
   Initiation, termination, interruption of computer tasks
4. **User guidance**
   Alerts, prompts, help messages, error messages
5. **Data transmission**
   Communication between users
   Communication between user and data or model subsystem
6. **Data protection**
   Protection against unauthorized access / computer failure

Dialog Desiderata

- **Minimize data entry**

  Intelligent defaults; Use previous run as template; Save standard user preferences; Import data when possible

- **Customize features**

  Frequently used commands at top level; Macro facility; Full/abbreviated menus; Verbosity of help

- **Don't overload memory**

  People can hold only about 7 distinct "chunks" in short-term memory. Organize displays so related information appears together.

- **Facilitate understanding of problem and model**

  Explore consequences of actions; Explore data; Check consistency; Perform sensitivity analysis.

- **Allow user to approach problem in intuitive way**
- **Alert user to possible biases in preferred approach**

Types of DSS
(relevant to dialog subsystem)

- Real-time / Non real-time
- Group / Individual
- Distributed / Local

Human Factors in HCI Design

- Human behavior is observable and human performance is measurable
- Human factors is the study of how human performance depends on system design
- Human factors engineering
  - Makes use of accumulated knowledge of human factors
  - Apply engineering design process to develop effective and efficient human/system interaction
- Maxims:
  - User should have to adapt as little as possible to interface
  - Interface should be designed to be easy and natural to learn

Usability

- *To users, the interface is the system*
- Some components of usability
  - Ease of learning
  - Efficient user task performance
  - Effective user task performance
  - Subjective user satisfaction
  - User retention over time
- Usability is measurable characteristic of a system
  - If we can measure it, we can improve it

Why Usability?

- **Costs**
  - Initial cost of system is paid once
  - Costs of lost productivity and error recovery are paid every time the system is used
  - Cost of user training and customer support can wipe out profits
- **Market forces**
  - Users will switch to competitors with more usable systems
- **Software engineering**
  - Traditionally the "important stuff" consisted of the system functions
  - Interface consumes an increasingly large share of project resources
  - Structured process of design for usability is essential for wise use of resources

How to Achieve Usability

- Follow structured engineering approach to user interface design
- Design to satisfy measurable usability goals
- Empirically evaluate interface with respect to usability goals
- Prototype and refine
- Follow established usability principles and guidelines

See references for several good books on usability engineering

Usability Specifications

- A usability specification is a clearly defined and measurable target for some attribute of value
- How to measure usability
  - Develop representative benchmark tasks
  - Identify what will be measured for each attribute of value
  - Develop targets for the measures
- Examples:
  - Not a specification: "System should be easy to use"
    - » Identifies the target but is not clearly defined or measurable)
  - Acceptable specification: "Responses to ease-of-use questions on user questionnaire must average at least 4 out on a scale of 1 to 5" is a specification
    - » The questionnaire is part of the specification
    - » The questions define usability targets

Behavioral and Constructional Domains

	Behavioral	Constructional
What is being developed	Interaction component of interface	Interface software (to support interaction)
What view is adopted	View of the user	View of the system
What is described	User actions, perceptions, and tasks	System actions in response to what the user does
What is involved	Human factors, scenarios, detailed representations, usability specifications, evaluation	Algorithms, callbacks, data structures, widgets, programming
The locale	Where interaction designers and evaluators do their work	Where interface software implementors do their work
The test	Procedures performed by the user	Procedures performed by the system

(Hicks and Hartson, 1993)

Interface Development: A Two-Part Process

- **Part One: Interaction component**
  - Look and feel of the system
  - Development occurs in behavioral domain
- **Part Two: Interface software**
  - How the look and feel is instantiated in software
  - Development occurs in constructional domain
- **These parts interact**
  - Inherent conflict: *better for the user may be harder for the programmer*
  - Available tools limit capabilities of interaction component
  - **The best toolkit can't rescue a bad design!**
- **Design must involve:**
  - User interaction developer
  - Interface software developer
  - Problem domain expert
  - Technical expert

Avoid "human factors as peanut butter" (spread over design after it is otherwise finished)

(Hicks and Hartson, 1993)

Standards and Guidelines

- User interaction standards
  - ISO 9241 - international user design standard (see "bluffer's guide" at http://www.userfocus.co.uk/articles/ISO9241.html)
  - Sect. 508 of Rehabilitation Act of 1973 - Standard for access to disabled
  - Standards need interpretation to be useful
- Public domain user interaction design guidelines
  - Published in trade books & published reports
  - Applying them in specific situations is more than common sense
- Commercial style guides
  - Description of interaction styles or objects
  - Guidance on when or how to use
  - May include illustrations, screen templates, toolkits
  - *Many have not had much professional human factors input!!*
- Customized style guides
  - Enterprise or project
  - Every project should have at least a simple style guide
  - Prevents inconsistencies, constant remaking of decisions
  - A few simple rules may suffice

Many useful links at http://www.usernomics.com/ergonomics-standards.html

Standards: Pro and Con

- Pro
  - Promote ease of learning and use
  - Assist in procurement, development, evaluation
  - Facilitate reuse
- Con
  - Standards inhibit innovation
  - Standards limit the ability to customize
  - Designers may substitute conformance to style guides for rigorous empirical usability testing

Guidelines for User-Centered Design

- Know the user
- Involve the user via participatory design
- Prevent user errors
  - "To err is human; forgive by design"
  - Prototyping helps identify common errors
- Optimize user operations
  - Most effect for least effort
  - Abbreviations; accelerator keys; macros
- Keep locus of control with user
  - e.g., "Enter next command" vs. "Ready for next command"
- Help the user get started with the system
  - User should need no more than one screenful of information to get started

Guidelines for Cognitive Issues

- Give user task-centered mental model of system
- Be consistent
  - Inconsistency: Macintosh “delete file” and “eject disk”
- Keep it simple
- Give user frequent closure on tasks to economize memory
- Let the user recognize rather than recall
- Use cognitive directness
  - Use “command-c” for cut, not “Esc-F7”
- Use familiar analogies
  - Drag and drop; folders on desktop; trash can

Guidelines on Feedback

- **Use informative feedback**
  - e.g., trash icon expands when item is dragged in
- **Give the user appropriate status indicators**
  - e.g., clock with moving hands indicates lengthy operation
- **Use user-centered wording in error messages**
  - "Operation failed - Error number 173" versus "There is not enough memory to open application. Try closing an application"
- **Use positive non-threatening wording in error messages**
  - e.g., NOT "catastrophic error, logged with operator"
- **Use specific, constructive terms**
  - "Illegal entry" versus "Inventory numbers range from 0000 to 9999"
- **Make the system take the blame**
  - "Illegal command" versus "Unrecognized command"

Other Interaction Guidelines

- Do not anthropomorphize; use humor cautiously
- Use modes cautiously
  - Same action should usually have same result
  - Use preemptive mode only when closure is necessary before continuing (e.g., "are you sure you want to delete?")
- Make actions easily reversible
  - Classic example of guideline that's nice for users and hard for programmers
  - What "undo" should mean is not always obvious
- Accommodate individual differences
  - Good predictors of performance: spatial visualization ability, vocabulary and logical reasoning ability
  - Give user control of interface (e.g. short and long menus)
- Accommodate user experience levels
  - Novice: no syntactic knowledge; little semantic knowledge
  - Intermittent user: rusty syntactic knowledge; good semantic knowledge
  - Power user: both semantic and syntactic knowledge

Display Guidelines

- Maintain display inertia
- Organize the screen to manage complexity
  - Minimize density
  - Group related information
  - Balance information and use plenty of white space
- Get user's attention judiciously
  - Only 2 levels of intensity on a single screen
  - Use underlining, bold, etc. judiciously
  - Use upper and lower case (takes less room and faster to read)
  - Use intense attention getters (blinking) only when necessary
  - Use of color:
    - » Be sparing on number of colors (no more than 4 colors)
    - » Should conform to user expectations (green = OK; yellow = warning; red = problem)
    - » Provide alternative coding for users with color deficiency

Typical User Complaints

Novice user

- The system didn't tell me what to do next.
- The system wouldn't let me go back and change my entry
- I'm not sure whether I made the correct entry.
- I don't understand what I did wrong.
- How do I change my response?
- I have too much trouble finding the right keys.
- I don't understand the error messages.
- I have to type so much.
- There are so many things to remember.

Expert user

- The system makes me go through too many procedures.
- I don't need a dissertation every time I make an error.
- The system keeps going back to the base menu.
- I have to keep typing in the same thing over and over again.
- Why can't I go from A to C without doing B every time?
- Why do I have to enter that information when I know it's in the database?

Interaction Modalities

1. **Menu selection**
   - easy to learn
   - power users may become impatient
   - good organization of menu hierarchy is essential
   - follow menu standards (e.g., commands on “file” menu”)
2. **Command language**
   - harder to learn (esp. nonintuitive or abbreviated commands)
   - power users like flexibility
3. **Forms**
   - intuitive
   - often tedious (but good use of defaults can help)
4. **Natural language**
   - most natural for inexperienced users
   - technology still cumbersome
5. **Hypertext**
   - intuitive
   - easy to get lost in a complex web of links
   - good site design & navigation guides are essential
6. **Direct manipulation**
   - easy to learn and use
   - harder to program (but toolkits help)
   - difficult to do in web-based applications

Windows Design Guidelines

- Don't overuse windows; minimize window manipulation
- Appearance and behavior of primary window should be consistent
- Use different windows for different independent tasks
- Use different windows for different coordinated views of same task

Menu Design Guidelines

- Organize hierarchical menus according to user tasks and system functions
- Use meaningful groupings and orderings of menu choices
  - Adhere to common practice when possible
- Use brief descriptions for menu choices
- Use consistent layout across all menus and keep screen uncluttered
- Allow shortcuts

Forms Design Guidelines

- Use consistent visually appealing layout and content
- Do not assume that existing paper forms convert directly to screen design and a good user interface
- Use appropriate visual cues for fields on forms
- Use familiar and consistent field labels and abbreviations
- Use logical navigation among fields
- Use logical navigation within fields
- Support editing and error correction of fields
- Use consistent, informative error messages
- Provide explanatory messages for expected field inputs
- Provide default values for fields whenever possible
- Provide a completion indicator on each form-filling screen

Box Design Guidelines

- Use brief but comprehensible instructions
- Use carefully worded messages
- Use logical groupings and orderings of objects in a box
- Use visual cues to delineate groupings within boxes
- Keep layout consistent and visually appealing
- Make defaults, such as a button choice, visually distinctive
- Menu selections that lead to dialogue boxes should contain a visual cue (e.g., ellipsis …)
- Boxes should disappear under user control

Typed-Command Language Design Guidelines

- Use a consistent rule of formation for entering commands
- Choose meaningful, specific, distinctive command names
- Apply consistent rules for abbreviating commands
- Allow easy correction of typing errors
- Allow frequent users to develop macros
- Provide auto-complete when possible

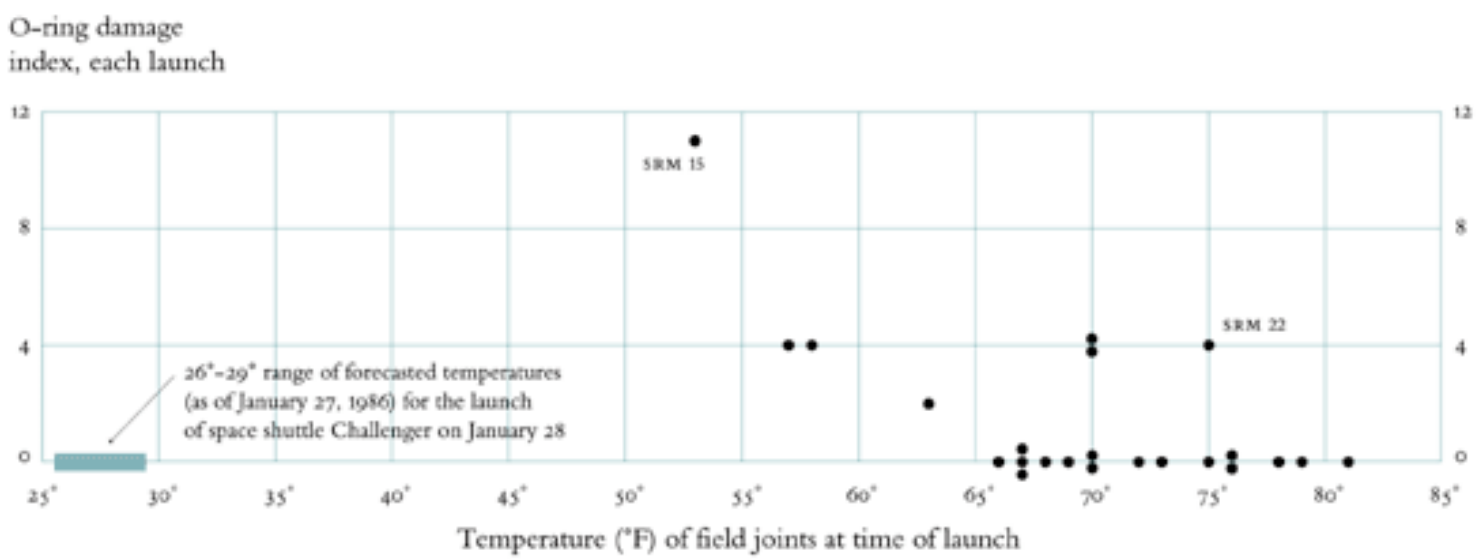
Graphical Interface Design Guidelines

- Use real-world analogies as much as possible
- Keep visual representation as simple as possible
- Show different views of the same visual object
- Use color sparingly and meaningfully
- Use video sparingly

Guidelines for Visual Display of Quantitative Information

(Tufte, 1983)

- **Maximize data ink** (percent of ink used to convey meaningful information)
- **Avoid chartjunk** (content-free decoration)
  - **Slideshow packages are notorious for chartjunk**
- **Show data variation, not design variation**
  - **Don't use areas to convey 1D information**
  - **Avoid lies due to scale changes**
- **Visual to verbal translation should be intuitive**



Cover photo, E. Tufte, *Visual and Statistical Thinking: Displays of Evidence for Making Decisions*

Input / Output Mode Examples

Input:

- Keyboard - alphanumeric
- Keyboard - cursor and function keys
- Mouse, trackball, trackpad
- Light pen
- Pressure pad
- Optical scanning
- Touch screen
- Voice activated
- Glance activated

Output:

- Text hardcopy
- Graphics hardcopy
- Character screen (monochrome/color)
- Graphics screen (monochrome/color)
- Sound
- Video
- Virtual environment immersion devices
  - Motion & acceleration simulator
  - Pressure simulators

Interface Levels

- Semantic: concepts; meaning
- Syntactic: rules of dialogue construction; grammar
- Lexical: mechanisms for specifying syntax; words, menu items, etc.
- Device: hardware

Aim for consistent, understandable straightforward interface at all levels

Intelligent Interface

- **Dialog Management System has model of user dialog preferences / needs**
- **Dialog adapts based on user, system state and world state**
- **Example determinants of user model:**
  - Expertise in problem domain
  - Expertise in system
  - Experience with computers
  - ”Cognitive style" (analytical/intuitive)
- **Example system/world characteristics affecting dialog:**
  - Time pressure
  - Cognitive workload
  - Precision required in result
- **Interaction between user mdoel and system/world model**
  - User familiarity with problem type

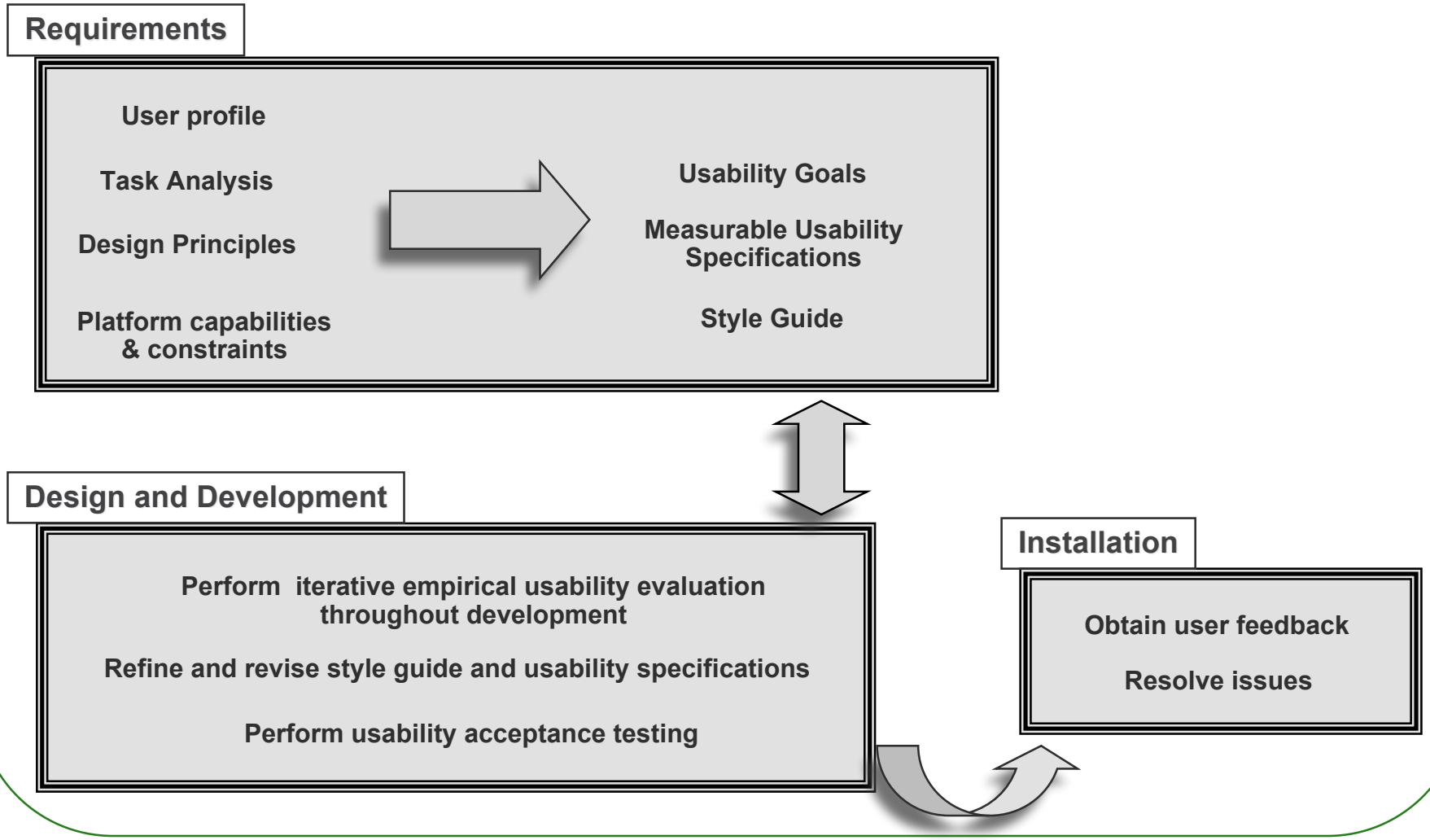
Intelligent Interface: Caveats

- Changing display formats and/or input modes in real time can be confusing (and dangerous)
- Customization (under control of user) achieves many of benefits of intelligently adapting interfaces without many of the dangers
- Avoid high-tech for high-tech's sake
- Actions taken by interface should be:
  - Obvious or explained to user
  - Easily grasped by user
  - Reversible by user with a simple toggle

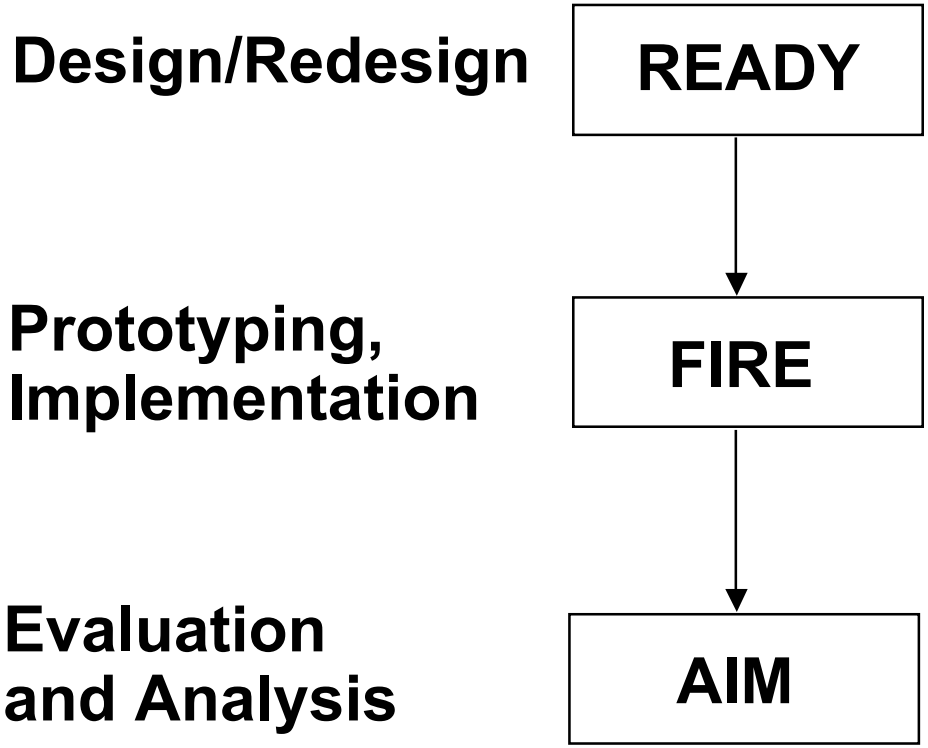
Interface Design: The Process

- Practice iterative, evaluation-centered design
- Inherent dilemma:
  - You can't evaluate an interaction design until after it is built
  - After building changes are difficult
- Solution: rapid prototyping
  - Representation - how are interaction designs represented in the prototype?
  - Scope - does the prototype include whole system or just the interface?
  - Executability - can the prototype be executed at any time?
  - Maturation - how does the prototype grow into a product?

Evaluation Centered Design for Usability



Artillery Approach to Interface Design



Prototyping the HCI: Storyboarding Environments

- Hypertext tools
- Interface building tools
- Prototyping environments
- Spreadsheets
- Draw packages and slideshow managers

Process of Interface Design

1. Perform task analysis
2. Identify input/output requirements for each task
3. Develop empirical measures of interface quality
4. Prototype interface design based on task analysis & input/output requirements. Use dimensions of interface quality as design goals.
5. Evaluate prototype using measures developed in Step 3.
   - Formative evaluation: evaluate to refine design
6. Repeat steps 3-5
7. Deliver final product
   - Summative evaluation: evaluate to assess quality of product

Attributes for HCI Evaluation

(subjectively measurable)

1. **Simple and natural dialog**
2. **Uses user's language**
3. **Minimizes user's memory load**
4. **Consistent**
5. **Provides feedback**
6. **Clearly marked exits**
7. **Shortcuts are available**
8. **Error messages**
   - understandable
   - provide suggested recovery strategies
9. **Prevents errors**
10. **Understandable intuitive displays (especially of complex quantitative data)**

Attributes for HCI Evaluation

(objectively measurable)

1. **Time to solution**
   - problem: type
   - problem: novel / familiar
   - user: inexperienced / experienced
2. **Requests for assistance**
3. **Number of errors**
4. **Time to recover from error**
5. **Quality of solution**
   - measures may be objective and/or subjective

Usability Evaluation

- Form hypothesis
- Design study with human participants
- Collect performance data
- Analyze data
- Confirm or refute hypothesis
- Evaluate impact of results on design decision

In Summary...

References

Bailey, Robert W. *Human Performance Engineering* Prentice-Hall, 1996.

Hicks, D. and Hartson, H.R. *Developing User Interfaces: Ensuring Usability Through Product and Process*. Wiley, 1993.

Kuniryski, M., *Observing the User Experience: A Practitioner's Guide to User Research*. Morgan Kaufmann, 2003.

Mayhew, D.J. *The Usability Engineering Lifecycle*. Morgan-Kaufmann, 1999.

Tufte, E.R. The Visual Display of Quantitative Information. Cheshire, CT: Graphics Press, 1983.

It's not about human/computer interfaces, but no reference list on human factors is complete without mentioning the Gettysburg Website: http://www.norvig.com/Gettysburg/