Chapter 2

TOPICS:

• Organizing usability issues: The User Action Framework

• Selected design guidelines, with examples

• Guidelines specifically for Web design

• Usability inspection

• In-class exercises on guidelines

Copyright © 2001 H. Rex Hartson and Deborah Hix.

All rights reserved. No part of this material may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, microfilming, recording, or otherwise, without prior written permission of its authors.
THE USER ACTION FRAMEWORK (UAF)

• The Interaction Cycle is highest level of categories in the UAF

* Adapted and extended Norman's "stages of action" model


• All about what users think, do, and see during cycle of interaction with computer

* THINK – Cognitive actions

* DO – Physical actions

* SEE – Perceptual actions
THE USER ACTION FRAMEWORK (UAF)

6.3 UAF Guides
THE USER ACTION FRAMEWORK

• Integrated framework for
  * Usability inspection
  * Usability problem reporting
  * Usability data management
  * Design guidelines
THE USER INTERACTION CYCLE

- Simplest view of the Interaction Cycle
THE USER INTERACTION CYCLE

• Simplest view of the Interaction Cycle
  * Planning (What to do)
  * Translation to determine actions (How to do it)
  * Physical actions (Doing it)
  * Assessment of outcome (Was it done right?)
SELECTED DESIGN GUIDELINES TO SUPPORT USER IN PLANNING

• Provide clear model of how users view system in terms of tasks
  Cartoon: monsters with clubs ->->->

• Make possibilities clear for what users can do at every point
  Tell users where they are and where they can go
  the king ->->->
  Blacksburg sign ->->->

• Keep users aware of task progress (what's been done and what's left to do)
SELECTED DESIGN GUIDELINES TO SUPPORT TRANSLATION

• Provide effective affordances – visual cues (e.g., in labels, data field formats, icons) that help users get access

* Help users determine what to do to get started

Powerpoint – getting started

* Help users predict outcome of actions

* Furnish defaults (e.g., most likely values, cursor position)

* Support human memory limits with recognition over recall

Simple example: menu over typed command (e.g., Unix, a good example of design for experienced user over new user.)

file naming w-o list in Save as

file naming w list in Save as
SELECTED DESIGN GUIDELINES TO SUPPORT TRANSLATION

* Accommodate different user classes with preferences

<table>
<thead>
<tr>
<th>Lead</th>
<th>Novice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Follow</td>
<td>Intermittent</td>
</tr>
<tr>
<td>Get out of the way</td>
<td>Expert</td>
</tr>
</tbody>
</table>

Cartoon: "This system's a snap" -> -> ->
SELECTED DESIGN GUIDELINES TO SUPPORT TRANSLATION

• Support user with effective presentation of visual cues

* Make noticeable
  - Object contrast, size, layout complexity, location with respect to user focus

* Make legible, readable (discernable)
  - Font size, font type, font color, font contrast

* Avoid irritation in presentation of cognitive affordances (e.g., color, blinking, audio, offensive messages)
SELECTED DESIGN GUIDELINES TO SUPPORT TRANSLATION

• Help user determine actions with effective **content/meaning** in visual cues

* Design for clarity
- Use *precise wording* in labels, menu titles, menu choices, icons, data fields
- Example: "Email" rather than "Answer Mail" or "Read Mail".

"Answer mail' is more specific, but is NOT more precise

- Be predictable; help users predict outcome of actions (feedforward)

- Be consistent

- Use consistent wording in labels for menus, buttons, icons, fields

  click retry to continue ->->->

- Use appropriate layout and grouping by function to convey content and meaning

options dbx - layout, grouping  -> options dbx - layout, grouping (solution) ->
SELECTED DESIGN GUIDELINES TO SUPPORT TRANSLATION

• Help user determine actions with effective content/meaning in visual cues

* Prevent translation errors

- "To err is human; forgive by design"

- Example: Gray out to make inappropriate choices unavailable (case of content conveyed by presentation medium)

- Example: Guide users with correct data entry (e.g., with data formats, field size, defaults, sample values)

  E.g., task series - date format missing --> -->
  (Task series is also not user-centered term)

* Help users recover from errors

- Provide clear way to undo (multiple levels) and reverse actions

- Offer helpful/constructive error messages
SELECTED DESIGN GUIDELINES TO SUPPORT TRANSLATION

- Support user with effective task structure and interaction control

* Keep locus of control with users

- Avoid feeling of loss of control (e.g., bossy attitude projected to users)
  
  E.g., "You need to answer your mail" or "Enter next command" vs. "Ready for next command". User should feel in charge.

- Avoid too much automation and real loss of control

- Example: Changing folder name "IRS" to "Irs"
  
  "user can't really want all caps"

  E.g., H. Rex Hartson problem, autoformat, locus of control > > >

  H. Rex Hartson problem, explanation > > >
SELECTED DESIGN GUIDELINES TO SUPPORT TRANSLATION

• Support user with effective **task structure and interaction control** (continued)

* Design task structure for preferences and efficiency

- Provide alternative ways to perform task
- Provide shortcuts (e.g., hot keys)
- Anticipate likely related tasks, support task thread continuity
- Provide (just) enough automation
- Support task thread continuity
- Example: If message suggests something, offer an easy way to do it

E.g., create Folder under Save As >>
SELECTED DESIGN GUIDELINES TO SUPPORT USER PHYSICAL ACTIONS

• Support user with effective design for **perception** of physical affordances – help in **seeing** objects to manipulate

* Make objects perceivable, legible, noticeable

• Support user with effective design for **manipulation** of physical affordances – help in **doing** actions

E.g., darn these hooves ->->->

* Avoid physical awkwardness

- Example: Time-consuming switches between multiple input devices (e.g., mouse and keyboard, touchscreen)

* Accommodate physical disabilities – limited motion, motor control, vision, hearing

* Locate related clickable objects close together but not too close
SELECTED DESIGN GUIDELINES TO SUPPORT USER ASSESSMENT OF OUTCOME

• Make sure of existence of feedback

* Provide feedback – no news is no news

Especially important for e-shopping

* Provide progress report on long operation (e.g., percent-done indicator)

Sometimes hard to do in client-server environment, but definitely can be done -- partly a software architecture issue.

* Request confirmation as a kind of intervening feedback, to prevent errors (especially for potentially destructive actions)

Confirmation msgs x3 -->-->
SELECTED DESIGN GUIDELINES TO SUPPORT USER ASSESSMENT OF OUTCOME

• Support user with effective presentation of feedback

* Make feedback noticeable

  Same as for translation, with this being the most important:

  - Locate within user focus
    E.g., message in status message box not in user focus

  - Make large enough size to see

  - Present feedback, error messages promptly
    Less than 2 seconds in general

  - Make feedback persistent (avoid "flashing")
    I.e., removing before users can process (not flashing as in blinking)
SELECTED DESIGN GUIDELINES TO SUPPORT USER ASSESSMENT OF OUTCOME

• Support user with effective **content/meaning** in feedback

  Just like for cognitive affordances of translation

* Design for clarity

  - Support clear understanding of outcome (system state change), so users can assess effect of actions

  - Give clear indication of error conditions to help users recognize errors

* Design for completeness

  - Provide enough feedback so users can be either confident their command worked or certain about why it didn't

    E.g., add record to database and get "Record added successfully". Else users take time to go back and retrieve it to see if it is there

* Help users understand what the real error is

  E.g., slash in file name, bad error msg->->->
SELECTED DESIGN GUIDELINES TO SUPPORT USER ASSESSMENT OF OUTCOME

• Support user with effective **content/meaning** in feedback (continued)

* Make system take blame for errors

* Use helpful, informative error messages, not "cute" unhelpful messages

  E.g., unsupported means unsupported ->->->

* Be consistent

  - Label outcome (e.g., title of new screen or dialogue box) consistently with starting point and action (e.g., button label or menu choice)

    madam - destination match departure - add to data entry ->->->

* Be positive to encourage; avoid violent, negative, or demeaning terms (e.g., avoid use of "illegal")

  illegal error, sent to prison ->->->

- Error messages can have great psychological impact on users

  Several cartoons, etc. ->->->
SELECTED DESIGN GUIDELINES
INDEPENDENT OF PLACE IN
INTERACTION CYCLE

• Use user-centered wording – language of user and work context in displays, messages, and prompts
  
  E.g., gobbledy-gook message
  
  printer out of paper (not an error)
  
  printer timeout msg

* Avoid anthropomorphism – attributing human characteristics to non-human objects
  
  Can be demeaning, patronizing, deceitful
  
  Don't anthropomorphize your computer; it doesn't like it.
  
  suspicions confirmed
  
  dialogue from math program

* Avoid poor attempts at humor
  
  - Easy to do badly
  
  - Easily misinterpreted
DESIGN GUIDELINES: CONCLUSIONS

• Be cautious; think and interpret guidelines
* In application, they can conflict and overlap
• Design by guidelines, not by politics or personal opinion

Jim Foley: "The only correct answer to any UI design question is: It depends".