User Interface Design
& Development

Lecture 8
Usability Heuristics

João Pedro Sousa
SWE 632
George Mason University

today

- know the user
- know the tasks
- design the interface
  - form fill guidelines
  - feedback
  - considerate UIs
    - undo
  - task vs. overhead

previously: pros & cons of dialogue styles
form fill

	pros	cons
menu selection	• recognize rather than remember • shortens learning • structures decision making, dialogues and error feedback	• menu explosion • screen real estate • may slow experienced users
form fill	• simplifies data entry • modest training • easy to provide contextual help	• screen real estate • may be burdensome on users • layout is critical for clarity
command language	• flexible and powerful • fast for experienced users	• remember rather than recognize • harder to design dialogues • harder to provide feedback & targeted error messages
	…	

forms
clarity

remember the "golden" rules

- make it easy
  - as before, choose the terminology taking personas into account & work on keeping consistent pay attention to abbrevs
  - provide defaults whenever possible
- make it clear
  - group related fields & use available space generously
    - carefully align the contents of fields
  - clearly indicate optional fields
  - give opportunity to review before committing data
- keep it real
  - provide online help/tooltips on each field
    - if not obvious, explain why the information is being asked e.g., race & sex
  - support going back and forth between parts keeping previously entered data

example
Mason dining services survey

- which clarity rules are followed, or not, here?

8. On a typical day when you are on campus around meal time, in which campus building are you in most often immediately prior to...

Lunch: ☐ *click here if you are off campus at lunch time*

Please select

if Other Building not in above list:

Dinner (up to 8pm): ☐ *click here if you are off campus at dinner time*

Please select

if Other Building not in above list:

Late Night (8pm or later): ☐ *click here if you are off campus after 8 pm*

Please select

if Other Building not in above list:

Previous Next

example
Mason dining services survey

- ... and here?

12. If the campus were to add food service in new locations, what location(s) on campus would you find most convenient to purchase food? *(Please select all that apply.)*

☐ Arlington Original Building ☐ Other (Please Specify)

☐ Hazel Hall ☐ Don't know/Unsure

☐ Truland

Previous Next

example
Blackboard

Open in New Window (YES must be selected)

Yes No

Email Announcement

Send a copy of this announcement immediately

Students are still notified of this announcement even if this option is not selected

example
from the industry

- why is this information being asked?
- "are you: No"?

today

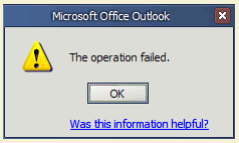
- know the user
- know the tasks
- design the interface
  - form fill guidelines
  - feedback
  - considerate UIs
    - undo
  - task vs. overhead

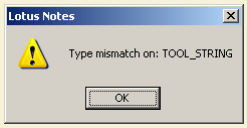
why feedback

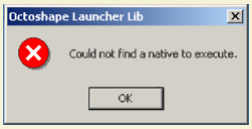
> The needs of both people and machines can be reconciled;
> users will respond more efficiently and intelligently
> if they receive meaningful feedback.
>
> B. Dwyer, Communications of the ACM, Sept. 1981

- messages should not be just pointers to manuals
  - e.g., Error 202
    if a user needs to read the manual to understand the problem,
    the message has failed
- when designing any kind of interaction, remember
  people quickly forget good experiences
  but remember the bad ones

feedback
negative examples from GUIs







more at
http://thedailywtf.com/Articles/Pop-up_Potpourri_0x3a__Perpetually_in_Beta.aspx

feedback
users shouldn't have to read a message twice

- be specific, clear, and concise
- be positive
- employ user's terminology
- design the delivery

- point out:
  - the source/field of the error
  - what went wrong
    enumerate all alternatives, if needed
  - suggest corrective actions

good & bad?

feedback

users shouldn't have to read a message twice

- be specific, clear, and concise
- be positive
- employ user's terminology
- design the delivery

fix

feedback

vague doesn't help

- be specific, clear, and concise
- be positive
- design the delivery
- employ user's terminology

examples

bad: vague	good: specific
Invalid input	The id number must be a 5-digit number
Invalid date	Format: MM/DD/YY
SYNTAX ERROR	Unmatched left parenthesis
Illegal entry	Type first letter: Send, Read, or Drop
Error accessing http://www.cs.gmu.edu: SOCKET: Connection refused	www.cs.gmu.edu is busy, try again later

discussion

- help doesn't

usability cannot be spread over a bad UI
like a layer of peanut butter
C. Lewis

feedback
don't blame users

- be specific, clear, and concise
- be positive
- design the delivery
- employ user's terminology

  - don't use hostile language
    - fatal error
    - execution aborted
    - illegal command
    - invalid entry
  - state what should be there, not what's wrong or missing

feedback
offer context-specific help

- be specific, clear, and concise
- **be positive**
- design the delivery
- employ user's terminology

same examples as before

negative	positive
Invalid input	The id number must be a 5-digit number
Invalid date	Format: MM/DD/YY
SYNTAX ERROR	Unmatched left parenthesis
Illegal entry	Type first letter: Send, Read, or Drop
Error accessing http://www.cs.gmu.edu: SOCKET: Connection refused	www.cs.gmu.edu is busy, try again later

feedback
offer context-specific help

- be specific, clear, and concise
- **be positive**
- design the delivery
- employ user's terminology

more examples

negative	positive
Illegal year!	The year must be between 1975 and 1995.
Your id # has characters!	The id # must be numeric. Please re-enter.
Part Code not numeric.	Part Code must be 3 numeric digits.

feedback
part of UI: system-initiated tasks

- be specific, clear, and concise
- be positive
- design the delivery
- employ user's terminology

  - design feedback with the same care: scenarios, etc.
  - provide timely feedback
    e.g., detect and react to situations ASAP, not at the end of operations
  - consider distractions vs. getting the user's attention
    - sounds/beeping
    - placement and highlighting
    - tradeoff between explanatory messages and the time it takes to read them

the kind of attention
you may expect from users
is less than ideal

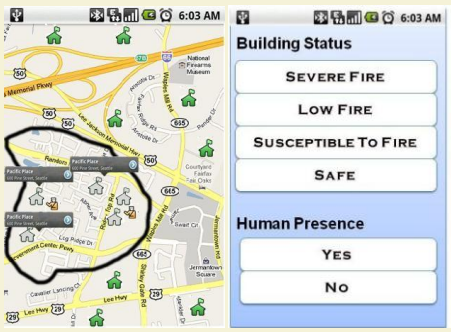
- in Germany
  http://www.youtube.com/watch?v=ONCrE4IoSsY
- in England
  http://www.youtube.com/watch?v=w_mkwB9ayK4&feature=related
- in the US
  http://www.youtube.com/watch?v=fJuNgBkloFE

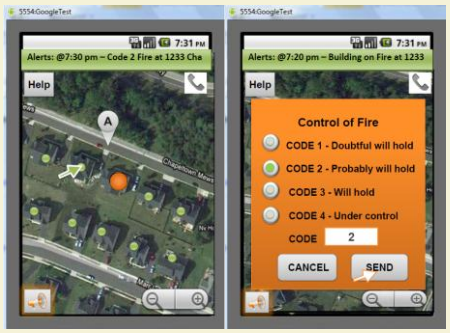
- good UI design minimizes the knowledge gap
  between users and the app

feedback
users have a lot on their heads

- be specific, clear, and concise
- be positive
- design the delivery
- employ user's terminology
  - study your application's target personae
  - remember:
    error messages are the wrong place
    to teach computer-related concepts to users
  - employ a consistent vocabulary of interaction

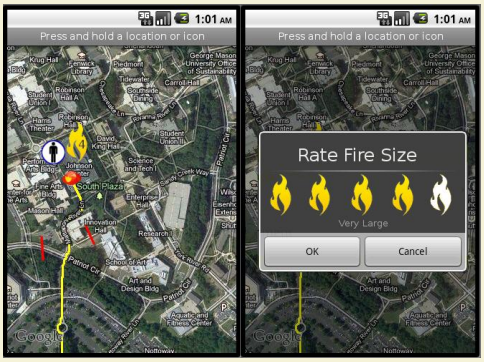
consistency
easier said than done





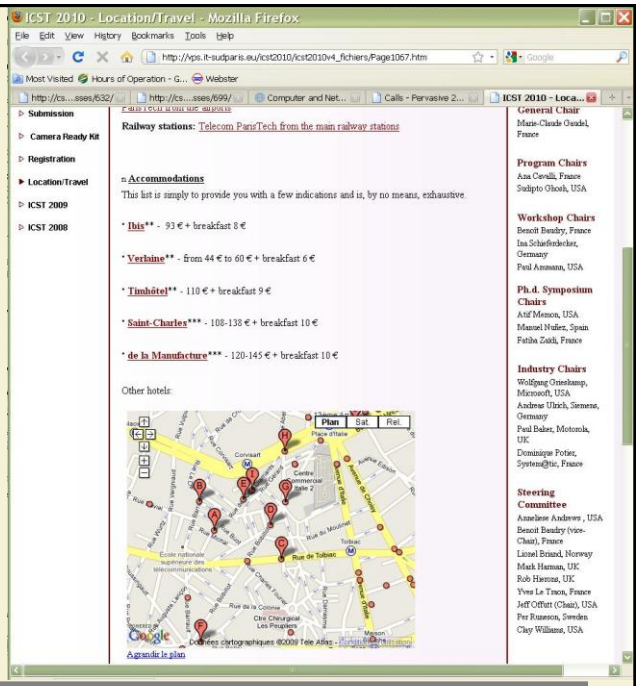
- consistency between input and output
  reduces cognitive load

consistency
easier said than done



- subtle differences are still differences

consistency
easy…



today

- know the user
- know the tasks
- design the interface
  - form fill guidelines
  - feedback
  - considerate UIs
    - undo
  - task vs. overhead

software is there to help the user not the other way around

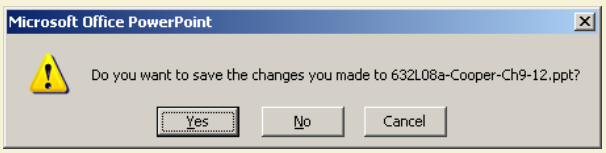
- people respond to computers as if they are sentient beings
  - software should be considerate, likeable, supportive
- designing considerate software is more work
- … but what's involved?

considerate UIs
never judge users

- don't tell users they're wrong
  the designer can never foresee all circumstances
  - give users choices, not orders
- be forthcoming
  - give users related information that might help them
    - wait time at printer queue
    - synonyms
    - highlight automatic spelling corrections
      …I will *defiantly* come to the meeting…
  - but, without getting on the user's way
    remember MS Clippy

considerate UIs
offer choices

- distinguish likely form unlikely choices
- make the consequences clear
  - "are you sure you want to quit? (*yes*, *no*, *cancel*)"
  - is *cancel* the same as *no*?



- hmmmm …. I didn't change anything …
  all I did was print!
- did I accidentally change something else?

considerate UIs
take responsibility

- a story from Windows XP
  about once a week the user shutdown his computer and went to bed…
  The next morning he would find a dialog box:
  "X is still running, would you like to close it?"
- the user's attention cycle may not be aligned with the app's cycle
  - users may get sidetracked and abandon the app or close it
  - users may close an app accidentally/push the power button
  - a failure may lead the app to stop (exception handling)
- allow users to complete part of the process now
  and come back later to finish
- design what should the app do when users are not available
  - some things can wait, sometimes waiting is the wrong decision

considerate UIs
take an interest

- remember what the user did /does often
  - computer memory/disk used to be scarce…
- form-centric apps
  might remember sets of field values entered in previous sessions
- document-centric apps
  might keep track of versions automatically
  - change management tools were developed in early 80's
  - negative example: MS office
- any app that manipulates/views data
  might keep track of previous user actions and support undo
  - negative ex: MS office threw away the undo stack with each save
  - negative ex: now they throw it away with each close

take 5

undo
has multiple purposes

- when users make mistakes,
  they often and rightly blame it on the computer or UI
  - *customer is always right*
  - UI should assume that everything users do is intentional
- undo has multiple purposes
  - exploring
    what does this function do?
  - hypothesizing
    does this function do X?

    beginners
  - and... rescuing mistakes

    experts
- original hypertext theorists thought
  back button would be used to fix navigation mistakes

make undo useful

- depth of undo
  - last operation
  - last few operations, one at a time
  - multiple operations at a time
    - show list of last operations and let user select
      e.g., MS office
    - more work to the UI designer, but users will thank you
- related operations
  - redo: undo the undo
  - repeat: apply same command to a different object

summary so far

- design system-initiated interactions inc. feedback as other interactions
- users have a lot on their minds
  and it's easy to catch them off guard
- considerate interfaces
  - minimize the knowledge gap
  - take responsibility
  - support *undo*

today

- know the user
- know the tasks
- design the interface
  - form fill guidelines
  - feedback
  - considerate UIs
    - undo
  - task vs. overhead

overhead aka excise

- two kinds of tasks
  - revenue: contribute directly to solve the problem
    e.g., design, code
  - overhead: must be done but are not really part of the problem
    e.g., compile, debug
- overhead tasks often
  satisfy the needs of the tools, not the users
- related to the accidental vs. essential in
  - Frederick Brooks, "No Silver Bullet: Essence and Accidents of Software Engineering," Computer, Vol. 20, No. 4 (April 1987) pp. 10-19
  - Aristotle's categories of nature
    http://plato.stanford.edu/entries/aristotle-metaphysics/

is the classification clear-cut?

what is overhead
depends on goals for task

- example: stopping at traffic lights while driving from A to B
  - you may argue that it's overhead because it slows the trip down
  - you might also argue that
    - it does not contribute to the goal of getting there fast
    - but it contributes to the goal of getting there safely
- how about software?
  - debugging
  - compiling
  - backing up

  when is that overhead, when is it not?

is overhead so relative we should drop the subject?

know your users
what is task for one persona may be overhead for another

- design interfaces that let each persona focus on their goals
  - you may have to provide different designs
- users are more productive when they can focus on task semantics
  - complex syntax is distracting
  - tradeoff with expressive power

design to accommodate skills in UIs

- novices need help
  - simple ways to accomplish limited tasks
  - explanation and guidance
- designing for novices slows down experienced users
  - features that help novices are overhead for everyone else!

design to accommodate skills in UIs

- novices need help
- designing for novices slows down experienced users
- design UIs
  - that make it easy to drop introductory features
  - make it easy to configure, customize, and aggregate features
  - provide undo instead of error prevention/hesitation
  - place windows carefully, don't make users move them each time
  - fewer separate windows with more expressive power/complexity
- alternatives
  - users configure UI
  - self-configuring UI

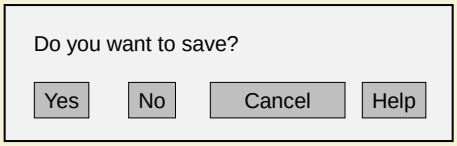
semantic UIs

- design for minimizing user input/commands
  - provide default behaviors and mechanisms to change it
  - make likely choices default, and unlikely choices available
  - separate commands from configuration, e.g. print vs. setup
- provide choices, minimize questions
  - toolbars offer choices, dialog boxes ask questions
- use dialogs to ask pressing questions
  - not to report normal/frequent behavior
- give information, not data
  - e.g., 40% saved vs. 20,000 bytes saved
  - if possible, indicate status visually, e.g. active/busy/idle

example
two design alternatives

consider an editor application with

- one Exit command followed by a dialog:



or

- one clearly visible Exit command which always saves
  - a second, more discrete, command *Quit-without-saving*

can you think of examples?

discussion
measuring usability

are these success criteria?

- time to perform a task, in seconds
- speed of user performance is the most important criteria
- users take less than 8 mouse clicks on average to perform task #5 with a standard deviation of less than 3
- the system issues a confirmation of receipt of the input data

summary

- overhead is accidental work that results of chosen/available technology and tools and UI design
- what constitutes overhead depends on
  - goals for the task
  - user persona
- given a set of goals and a user persona there are still many possible UI designs
  - evaluate interaction length and amount of data entry
  - evaluate separation of concerns on each screen & navigation
  - designers need to work hard to reduce overhead for users

discussion
usability is process

usability engineering
is intertwined with design

usability lifecycle

- pre-design
  - model the user, context & tasks
- design
  - participatory design: paratypes, prototypes, Wizard of Oz
  - analysis of current practice and competition
  - coordinated design & guidelines
- post-implementation
  - functional testing
  - empirical studies: lab, in situ, in the wild
- revise design for future releases

evaluation

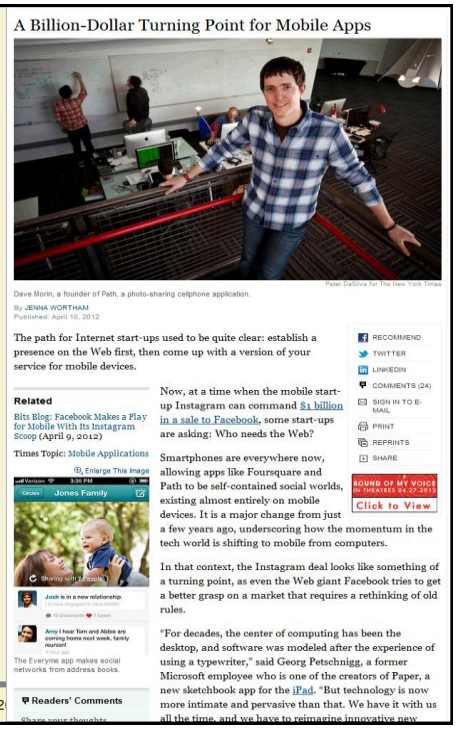
are you prepared?

SWE 632 covered 3 layers of expertise

	demonstrate
• engineer usability	project
• guidelines and best practices	exams
• code	labs

looking ahead

- mobile UIs
- interfaces embedded in the physical world
  - http://www.youtube.com/watch?v=2lXh2n0aPyw



A Billion-Dollar Turning Point for Mobile Apps

Dave Morin, a founder of Path, a photo-sharing cellphone application.

Peter DaSilva for The New York Times

By JENNA WORTHAM
Published: April 10, 2012

The path for Internet start-ups used to be quite clear: establish a presence on the Web first, then come up with a version of your service for mobile devices.

Now, at a time when the mobile start-up Instagram can command $1 billion in a sale to Facebook, some start-ups are asking: Who needs the Web?

Smartphones are everywhere now, allowing apps like Foursquare and Path to be self-contained social worlds, existing almost entirely on mobile devices. It is a major change from just a few years ago, underscoring how the momentum in the tech world is shifting to mobile from computers.

In that context, the Instagram deal looks like something of a turning point, as even the Web giant Facebook tries to get a better grasp on a market that requires a rethinking of old rules.

"For decades, the center of computing has been the desktop, and software was modeled after the experience of using a typewriter," said Georg Petschnigg, a former Microsoft employee who is one of the creators of Paper, a new sketchbook app for the iPad. "But technology is now more intimate and pervasive than that. We have it with us all the time, and we have to reimagine innovative new

Related

Bits Blog: Facebook Makes a Play for Mobile With Its Instagram Scoop (April 9, 2012)

Times Topic: Mobile Applications

Enlarge This Image

Jones Family

Sharing with 3 people

Josh is in a new relationship.

Amy I hear Tom and Abbie are coming home next week, family reunion!

The Everyme app makes social networks from address books.

Readers' Comments

RECOMMEND
TWITTER
LINKEDIN
COMMENTS (24)
SIGN IN TO E-MAIL
PRINT
REPRINTS
SHARE

SOUND OF MY VOICE IN THEATRES 04.27.2012
Click to View