

User Interface Design & Development

Lecture 6 Design

João Pedro Sousa

SWE 632
George Mason University

previously outside-in design

in a nutshell:

- know the user
- know the tasks
- design the interface



previously model user, context & tasks

what's relevant for the tasks

- knowledge
 - task semantics, computer semantics, app syntax
- goals
 - priorities, commitment, attention, responsibilities
- skills & perceptions
 - short & long-term memory, graphical interpretation, language speaking/understanding, visual impairment, dexterity...

previously model user, context & tasks

pre-design

- model user personas
- scenarios of use
- task models
 - functional requirement, decomposition
 - context: setting & concurrent activities, interruptions...
 - success criteria: learn, retention, speed, errors
 - frequencies for each persona
- leave for participatory design
 - identify candidate dialogue styles for each task x persona
 - dialogues - realization of tasks over a concrete syntax

today

outside-in design

- know the user
- know the tasks
- design the interface
 - purposes of interaction
 - dialogues
 - find info
 - enter data
 - app posture



broad purposes of interaction

- app-specific dialogue
 - e.g. pay the electric bill
 - set the home's security system to let the dog sitter come in at 1pm
- know/find something
 - read the news
 - best Korean restaurant in town
 - where to pay the electric bill
 - analyze data
- enter data
 - e.g. enter customer no. and credit card data
- others?

these kinds of interaction may appear separately
or together in some combination

several dialogue styles

	pros	cons
menu selection	<ul style="list-style-type: none"> • recognize rather than remember • shortens learning • structures decision making, dialogues and error feedback 	<ul style="list-style-type: none"> • menu explosion • screen real estate • may slow experienced users
command language	<ul style="list-style-type: none"> • flexible and powerful • fast for experienced users 	<ul style="list-style-type: none"> • remember rather than recognize • harder to design dialogues (chains of commands) • harder to provide feedback & targeted error messages
natural language	<ul style="list-style-type: none"> • best for untrained users • best flexibility and power 	<ul style="list-style-type: none"> • hard to manage context • ambiguous • complex, high-cost development

several dialogue styles

	pros	cons
form fill	<ul style="list-style-type: none"> • simplifies data entry • modest training • easy to provide contextual help 	<ul style="list-style-type: none"> • screen real estate • may be burdensome on users • layout is critical for clarity
direct manipulation	<ul style="list-style-type: none"> • presents task concepts visually • recognize rather than remember • good retention • encourages exploration • easier to avoid/recognize errors 	<ul style="list-style-type: none"> • may be hard to design & program • may become complex to use for all but simple cases

dialogues

rules of thumb aka golden rules

make it easy

- universal usability
 - identify the different user profiles and design for them
 - design UIs that adapt to increasing user experience
- reduce short term memory load
 - computers are good at remembering, people are not: remember that
- consistency
 - one concept - one syntactic representation
 - work on it: harder to get right than it sounds

dialogues

rules of thumb aka golden rules

make it clear

- offer feedback
 - not just for errors
 - more feedback for
 - less frequent operations
 - operations with larger consequences
- yield closure
 - design clear endings for each dialogue/task
 - e.g., confirmation of transaction in e-commerce

future class

see next slide

example

which rules are not being observed?

The screenshot shows the ePAR system interface. The header is green with the text "ePAR - The Volgenau School of Information Technology and Engineering". Below the header is a navigation bar with links: Home > ePAR, Reports, Period Selection, Preview, Print/Save, Submit PAR, and Help. The "Submit PAR" link is highlighted. On the left is a sidebar menu for "jsousa" with a tree view containing: ePAR (expanded), General, Research, Teaching, Service, Reports, FAQ, Feedback, My account, and Log out. The main content area shows a message box with two bullet points: "PAR submission successful." and "Thank you for using the ePAR system. We would appreciate it if you would take a few moments of your time to provide us with suggestions or Comments about your experience using ePAR. The following link will take you to our feedback page: ePAR Feedback". Below the message box is a text input field with the placeholder text "If you wish to submit your PAR for review, enter 'Yes' in the field below (without quotation marks) and press Submit. You may submit your PAR as many times as you wish until 23:59 on Friday, October 9, 2009." and a "Submit" button. At the bottom is a "Submission History" section with the text "This PAR was submitted for evaluation on: 2009-10-07 at 14:15:17." The footer contains the text "SWE 632 - UI design", "© Sousa 2012", and "Lecture 6 - Design - 11".

dialogues

rules of thumb aka golden rules

make it real

• errors

- people make mistakes: design for it
- instead of generic messages, pinpoint what went bad
 - more work to implement, but users will thank you
- recover as much state as possible
 - only ask the user to redo what was wrong
- the closer the syntactic flow is to the semantics of the task the less mistakes

• easy reversal aka undo

- if op can't be undone, explain consequences beforehand
- ≠ asking "are you sure...?" aka hesitation

dialogues

rules of thumb aka golden rules

the big tradeoff

- prevent errors
 - e.g., do not offer choices that should not be taken
- put the user in charge
 - don't try to make the software smarter than the user
 - in the least, users get annoyed, in the worst, they stop using it
 - remember: real experts know best than any smarts that can be designed into the software
 - design software that is flexible and accommodating of users
 - planes have crashed because the auto-pilot software refused to let the pilots do their job

dialogues

rules of thumb aka golden rules

the big tradeoff

- prevent errors
- put the user in charge

how to address the tradeoff?

- one possibility: leverage user profiling and authentication
 - novices get more prevention
 - certified experts get more freedom
- domain-specific and application-specific solutions
 - if operation is easily undone, don't bother with prevention/hesitation

outline broad purposes of interaction

- app-specific dialogue
- know/find something
 - read the news
 - best Korean restaurant in town
 - where to pay the electric bill
 - analyze data
- enter data
 - e.g. enter customer no. and credit card data
- others?

these kinds of interaction may appear separately
or together in some combination

SWE 632 - UI design

© Sousa 2012

Lecture 6 - Design - 15

finding things what are users looking for?

The screenshot shows a flight tracking website with several sections:

- Raw METAR**: METAR (KJFK) 10.0 kts, 347° (NW), 5000 ft. Full Report - FourPages
- Self Serve Advertising!**: Promote your business to your local community on underground.com. Try us today with a free \$50 credit! [Click to learn more](#)
- Nearby Airports**: Table with columns: City, Temp., Conditions, Updated.
- Flight Tracker Arrivals**: Table with columns: Flight, From, To, ETA, Status.
- History & Almanac**: Max Temperature, Min Temperature, Record (KJFK), Yesterday, Yesterday's Heating Degree Days: 13.
- Air Quality Forecast for Northern Virginia**: Table with columns: Air Quality, AQ Index, Pollutant.
- Best of Today**: Partly sunny. Highs around 60. Northwest winds 10 to 15 mph with gusts up to 25 mph.
- Tonight**: Mostly cloudy. Lows in the lower 40s. West winds 5 to 10 mph.
- Wednesday**: Partly sunny in the morning, then becoming mostly cloudy. A 20 percent chance of showers. Highs in the lower 60s. West winds around 5 mph.
- Wednesday Night**: Mostly cloudy in the evening, then becoming partly cloudy. Lows in the upper 40s. West winds around 5 mph.
- Thursday**: Mostly sunny. Highs around 70. West winds 10 to 15 mph.
- Thursday Night**: Mostly clear. Lows in the upper 40s.
- Friday**: Sunny. Highs in the lower 70s.
- Friday Night**: Clear. Lows in the upper 40s.
- Saturday**: Sunny. Highs in the mid 70s.
- Saturday Night**: Mostly clear. Lows in the lower 50s.
- Sunday**: Sunny. Highs in the lower 70s.
- Sunday Night**: Clear. Lows in the upper 40s.
- Columbus Day**: Sunny. Highs in the mid 70s.

look at
scrollbar

SWE 632 - UI design

© Sousa 2012

Lecture 6 - Design - 16

finding things less is better

WHAT DESIGNERS BUILD...



WHAT USERS SEE...



I want to
buy a
ticket.

How do I
check my
frequent
flyer miles?

users don't read they skim

what are the
important/frequent tasks?
for the kind of user

SWE 632 - UI design

© Sousa 2012

Lecture 6 - Design - 17

finding things keep it consistent and uncluttered

- keep consistency of
 - layout of screens
 - terminology: task-related keywords, titles, abbrevs
 - get someone other than the designer to review the consistency
- make it easy to assimilate
 - keep layout uncluttered
 - visually group related concepts, separate what is separate
 - use careful layout and spacing
 - highlighting
 - ...see next slide

SWE 632 - UI design

© Sousa 2012

Lecture 6 - Design - 18

finding things uncluttered highlights

- typical techniques
 - *italicized*, underlined, **bold**, **size**, **color**, alignment
 - sound
 - tactile, e.g. vibration
 - ambient light
 - smell...
- use an appropriate technique according to
 - size of the display
 - where is the user attention
 - milder if attention is on display, e.g. embedded on text that the user is reading
 - stronger if needs to attract attention
 - consider the impact of distracting the user (e.g., nasty beep while driving)

finding things uncluttered highlights

- use highlighting sparingly
 - if everything is highlighted then...
nothing is highlighted
the Times Square effect



highlight normally looks stronger to the user than to the designer

finding things reduce memory load and make it flexible

- reduce memory load
 - confine each task to one, or a small number of screens
 - don't make the user remember things
 - keep necessary/relevant information from one screen to the next (e.g., look at patriot)
- make it flexible
whenever relevant, allow:
 - sorting
 - showing/hiding of different fields
 - export into different formats (e.g. look at file explorer, email client)

discussion

- users are not designers
- designers are not users
- vice presidents are not designers
- less is more

outline broad purposes of interaction

- app-specific dialogue
- know/find something
 - read the news
 - best Korean restaurant in town
 - where to pay the electric bill
 - analyze data
- enter data
 - e.g. enter customer no. and credit card data
- others?

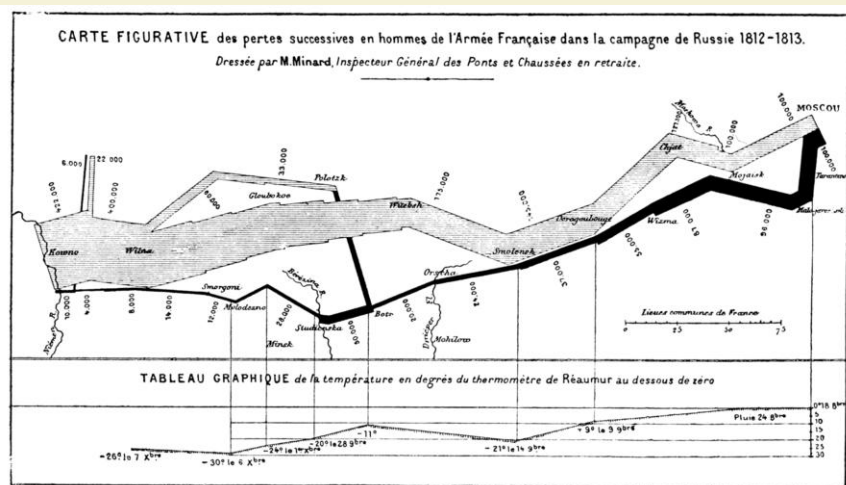
these kinds of interaction may appear separately
or together in some combination

SWE 632 - UI design

© Sousa 2012

Lecture 6 - Design - 23

presenting information *clear thinking made visible* Edward Tufte

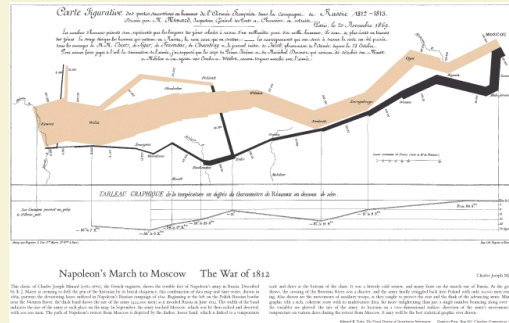


SWE 632 - UI design

© Sousa 2012

Lecture 6 - Design - 24

visual information design is hard



- hard to display multidimensional information on 2-D paper or screen
- screens have lower resolution and color discrimination than high-quality printing on paper

seven principles of visual information design

- enforce visual comparisons
- show causality
- show multiple variables
- integrate text, graphics, and data into one display
- ensure the quality, relevance and integrity of the content
- show things adjacently in space, not stacked in time
- do not de-quantify quantifiable data

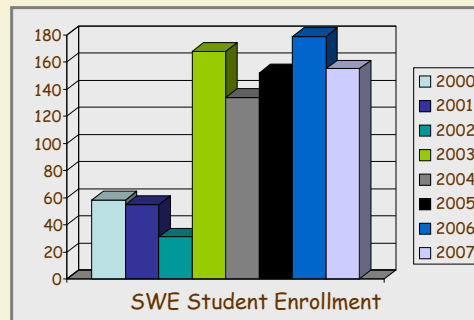
in order:

seven principles

1. enforce visual comparisons

- quantitative data is much easier to analyze visually
 - trends
 - before and after scenarios
 - related variables: remember Minard's

2000	58
2001	55
2002	31
2003	168
2004	134
2005	142
2006	179
2007	155



SWE 632 - UI design

© Sousa 2012

Lecture 6 - Design - 27

seven principles

2. show causality

- causality: relation between cause and effect
- Three Mile Island 1979
 - a valve was open that needed to be closed
red indicated closed, green indicated open
 - the operator saw green, thought "okay", and didn't close the valve for several crucial seconds...
 - current visualizations show two diff pictures for open/close
- Challenger disaster 1986
 - O-rings failed, causing the shuttle to blow up on launch
 - "...dangers of launching on such a cold day..."
 - "...had failed to adequately report these technical concerns to their superiors."
 - better charts could have clarified the relation between environment temperature and O-ring failure

SWE 632 - UI design

© Sousa 2012

Lecture 6 - Design - 28

seven principles

3. show multiple variables

- if information is affected by multiple variables they should all be visible
 - use overlays
- allow users see/hide some of the variables

4. integrate text, graphics, and data

- graphics give an overview, text and data add precision
- work at showing the link between all elements
 - make it easier for users to interpret

seven principles

5. ensure the quality, relevance and integrity

- don't show data just because it's available
incorrect or unneeded data:
 - slows users down
 - damages their trust in the application and in you
 - increases errors on the part of users
- consider the user's goals
 - remember that diff personae may have diff goals
- display only the information that supports those goals

seven principles

6. show things adjacently in space not stacked in time

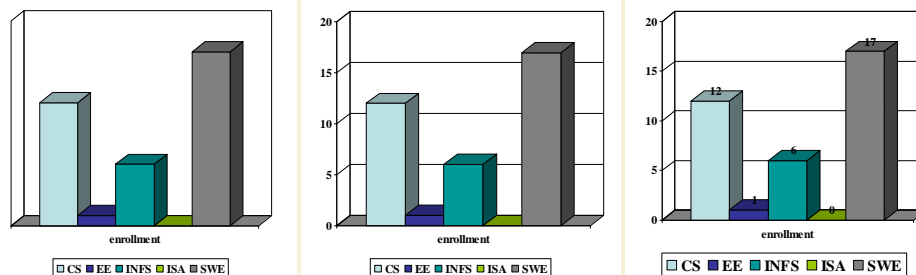
- for an overview,
a graphical representation of time
may work better than an animation

Project presentations	11/03	Direct Manipulation
	11/16	Usability Heuristics
	11/23	Hypertext
	11/30	Direct Manipulation
	12/07	Project presentations

seven principles

7. do not de-quantify quantifiable data

- charts are good, numbers add preciseness



take 5

outline broad purposes of interaction

- app-specific dialogue
- know/find something
 - read the news
 - best Korean restaurant in town
 - where to pay the electric bill
 - analyze data
- enter data
 - e.g. enter customer no. and credit card data
- others?

these kinds of interaction may appear separately
or together in some combination

enter data more rules of thumb

- be consistent
 - accept input that looks like the output
 - e.g., if you show a number as xxx-xx-xxxx
don't ask the user to enter it without the dashes
- be flexible
 - if there is no ambiguity, accept a variety of formats
- make it easy
 - infer as much as possible and allow user to change
 - e.g., shipping address same as billing address
 - minimize device switching
 - e.g., mouse for selecting, keyboard for entry
 - don't go overboard with error prevention

mm-dd-yy
mm-dd-yyyy
mm/dd/yy
mm, dd yyyy

see next slide

SWE 632 - UI design

© Sousa 2012

Lecture 6 - Design - 35

going over board with error prevention *look Mom, I can code a combobox!*

- e.g., select year of birth

year: 1900
1901
1902
1903
1904
1905
1906
1907
1908

- **Mason Dining Services Survey**
Now, thinking about those same meal purchases on a typical Sunday, for each meal and/or snack you purchase (on- or off-campus) using cash or a credit card, indicate the approximate amount that you spend for each meal and/or snack

Afternoon Snack:
\$3.00-\$3.49

Dinner:
\$5.50-\$5.99
Please select
Does not apply
Less than \$1.00
\$1.00-\$1.49
\$1.50-\$1.99
\$2.00-\$2.49
\$2.50-\$2.99
\$3.00-\$3.49
\$3.50-\$3.99
\$4.00-\$4.49
\$4.50-\$4.99
\$5.00-\$5.49
\$5.50-\$5.99
\$6.00-\$6.49
\$6.50-\$6.99
\$7.00-\$7.49
\$7.50-\$7.99
\$8.00-\$8.49
\$8.50-\$8.99
\$9.00-\$9.49

SWE 632 - UI design

© Sousa 2012

Lecture 6 - Design - 36

emerging technologies enter data & commands

- wiigee
 - <http://www.youtube.com/watch?v=Vb2tOOiwEhY&noredirect=1>
- kinect
 - <http://www.youtube.com/watch?v=yLqSY07tS64&feature=related>
 - <http://www.youtube.com/watch?v=v0G0Cb7Zqjk&feature=related>
- inertial sensor over bluetooth
 - <http://www.youtube.com/watch?v=I3HSFIw6V-k>
- tutorial for Android
 - <http://stackoverflow.com/questions/937313/android-basic-gesture-detection>

today outside-in design

- know the user
- know the tasks
- design the interface
 - purposes of interaction
 - dialogues
 - find
 - present data
 - enter data
 - app posture



posture: set of assumptions and attitudes

- UIs, and the underlying apps, may adopt different postures relative to users [Cooper+ 2007]
 - sovereign
 - transient
 - daemonic
- each of these leads to
 - different types of user interaction
 - different sets of behavioral attributes

sovereign posture

assumptions

- the only program running: has all of user's attention
- users quickly become experienced intermediates
 - interaction design optimized for
 - easy learning, and expressiveness

characteristics

- takes a significant part of the screen
- fairly passive: users take the initiative
- runs for many minutes, or hours
 - e.g., office apps: word processors, slide editors, spreadsheets...
- provide rich visual environment and feedback
- often support several dialogue styles, e.g. menus, shortcuts...
- often document-centric: users focus on a specific document

transient posture

assumptions

- the user will invoke it when needed: app does the job, disappears
- intermittent users: not the main focus of attention

characteristics

- applications with focused set of functionality
 - e.g., e-mail, file manager, media players
- use relatively small portion of the screen
 - one window or functional widget, e.g. windows media player

guidelines

- instructions must be available and clear for infrequent users
- make windows movable
so not to occlude sovereign apps
- remember the state ...next slide

transient apps ideally, learn user's preferences

- design the UI to remember what the user
 - does often
 - did last time
- negative examples
 - forget in which directories files are normally kept/saved or require directory customizations deeply buried in the interface
 - forget the printing setting for documents
 - for printing PowerPoint handouts, user needs to select save as pdf, handouts, 2-slides, gray scale, each time
 - ATM that always offers withdrawal amounts in multiple of \$20, even if user always withdraws \$50

daemonic posture

assumptions

- minimal interactions with the user, if at all

characteristics

- run in the background, mostly out of user's sight
 - e.g., printer drivers, network drivers

guidelines

- stay out of user's sight, unless strictly needed
 - negative example: a small-volume printer that pops up a little dialog box each time to inform it's printing
- control panels may be necessary to configure the app's behavior
 - keep them simple: remember to stay out of user's way

example of postures on the Web

- **content sites:** present information, little interaction
 - purpose: complete and detailed information, e.g. school site
 - design for: sovereign, emphasis on navigation
 - purpose: focused interest and occasional use, e.g. wikipedia
 - design for: transient, emphasis on search
- **transaction sites:** ecommerce, banking, travel, taxes...
 - normally sovereign; depending on app, may rely mostly on form fill, search, or navigation
- **Web apps:** downloaded and run under the browser process
 - goal: provide rich functionality without explicit & permanent installation of software: e.g. webmail, wikis, games

next slide

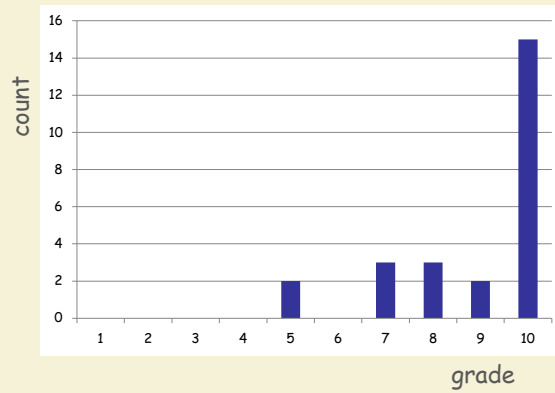
Web apps

- **sovereign: designed as desktop apps**
 - include multiple control and data objects
 - use panes to group functions
 - emphasize interaction as opposed to navigation
 - few or no request/response dialogs
- **transient: small, well-defined feature**
 - e.g. google search bar, cnn weather and QuickVote boxes
 - within a web page or embedded on the browser controls
 - quick occasional use
 - make login or data retrieval as simple as possible
- **note: internet-enabled apps \neq web apps**
 - use the internet, but don't run inside a browser, e.g. media players

discussion postures in Android?

- **normal interactions**
 - activities, e.g. enter a note in NotePad
- **services**
 - e.g. a music playing service

feedback on e1



- examples