

Human-Computer Interaction

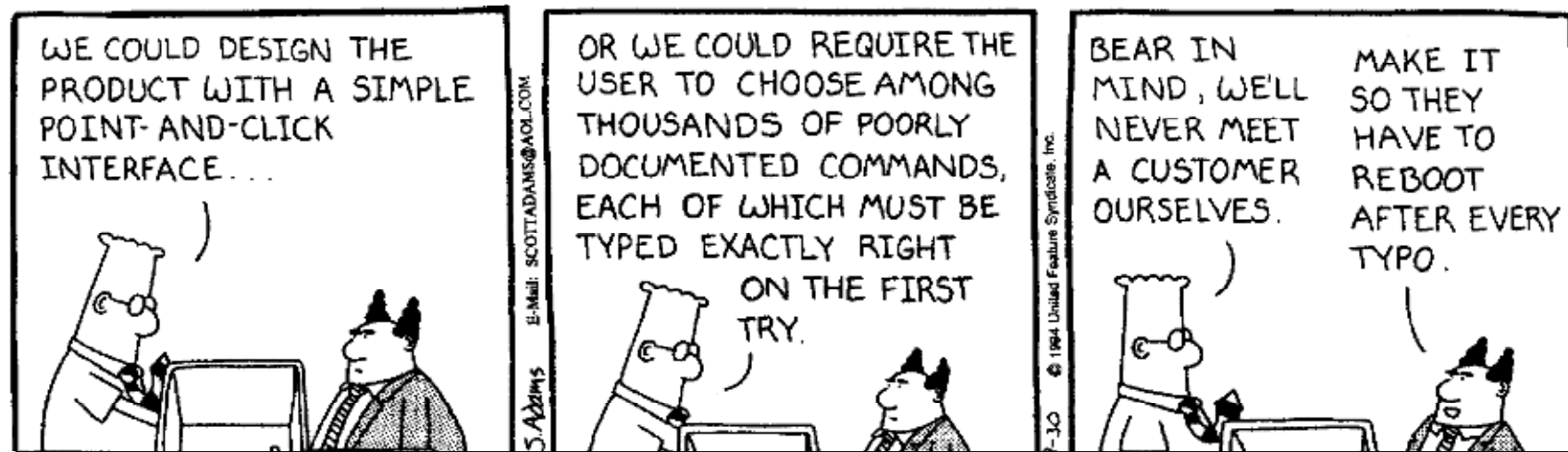
Termin 6:
Designing Interactive Systems
Principles & guidelines

readings:

Shneiderman: "Designing the User Interface", chapt. 2

Dix et al.: „Human Computer Interaction“, chapt. 5+7

how to design an interactive system?



key features of interactive systems

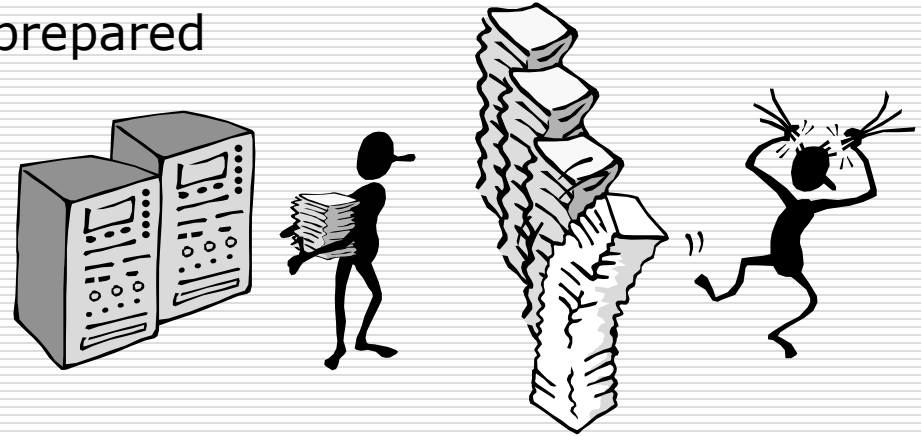
- interactivity
- accessibility
- acceptability
- usability
- engagement



interactivity

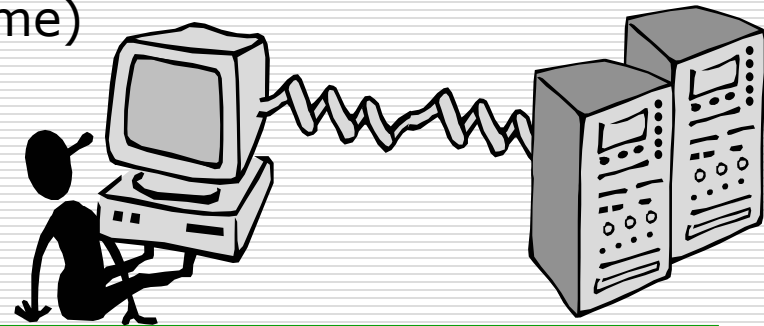
long ago, in a galaxy far away ... *batch* processing

- card stacks or large data files prepared
- long wait
- line printer output
... and if it is not right ...



now, computing is (should be) interactive

- rapid feedback
- the user in control (most of the time)
- doing rather than thinking



accessibility & acceptability

□ accessibility

- computers are everywhere
- always accessible for everybody?
- physical, conceptual, economical, cultural, social

□ acceptability

- does the system fit into people's life?
- politically acceptable, convenient, cultural and social habits, useful (beyond usable, useful in context?), economic



usability

□ definitions

- Shackel (1990): *Easy to use, easy to learn, flexible & engender good attitude in people*
- now mostly *efficiency and effectiveness*

□ a *usable* system:

- efficient, people can do things with appropriate effort
- effective, people can do things at all
- easy to learn & to remember
- safe to operate in all possible contexts
- high utility, does things that people want get done



engagement

- qualities that pull people in
- when a system is accessible, usable, and acceptable, engagement concerns with all qualities that make it memorable, satisfying, enjoyable and rewarding
- several key elements (Shedroff, 2001):
 - identity & authenticity (are you a „Mac person“?)
 - adaptivity with change of difficulty, pace, movement
 - narrative (tells a good story)
 - immersion (gives feeling of being wholly involved)
 - flow (gradual change between states, keeps you in)



the main goal of interaction design

designing for maximum **usability**

there can be other secondary goals, just as important

- engagement and fun, e.g. in computer games
- ...



how to support usability?

1. Learnability

- the ease with which new users can begin effective interaction and achieve maximal performance

2. Flexibility

- the multiplicity of ways the user and system exchange information

3. Robustness

- the level of support provided the user in determining successful achievement and assessment of goal-directed behaviour



1. Learnability

- predictability
 - determining effect of *future* actions based on past interactions and operation visibility
- synthesizability
 - assessing the effect of *past* actions
- familiarity
 - how prior knowledge applies to new system
- generalizability
 - extending specific knowledge to new situations
- consistency
 - likeness in input/output behavior arising from similar situations or task objectives



2. Flexibility

- dialog initiative
 - freedom from system-imposed constraints on input dialog
- multithreading
 - ability to support user interaction for multiple tasks
 - concurrent vs. interleaving; multimodality
- task migratability
 - passing responsibility for task execution between user and system
- substitutivity
 - equivalent values of input/output can be substituted
 - representation multiplicity
- customizability
 - interface can be modified by user or system



3. Robustness

- observability
 - ability of user to evaluate the internal state of the system from its perceivable representation
- recoverability
 - ability of user to take corrective action for errors
 - reachability; forward/backward recovery; commensurate effort (effort of undo = effort of do)
- responsiveness
 - perceived rate of communication with the system
 - stability
- task conformance
 - degree to which system supports the user's tasks



the design process

designing for maximum usability

designing interactive systems...

design *interactions* not just interfaces

not just the “immediate” interaction

e.g. stapler in office – technology changes interaction style

- manual: write, print, staple, write, print, staple, ...
- electric: write, print, write, print, ..., staple

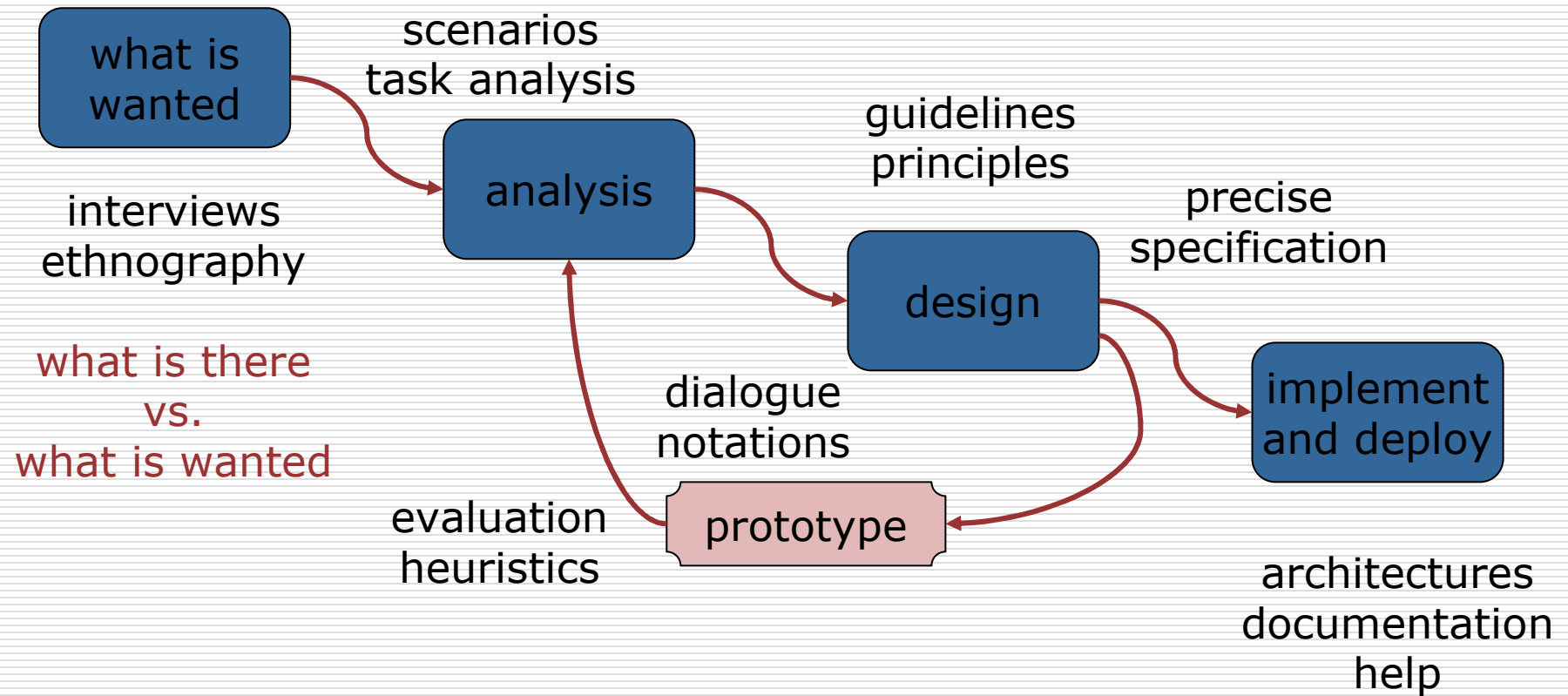
designing *interventions* not just artefacts

not just the system, but also ...

- documentation, manuals, tutorials
- what we see, read, say, do, etc.

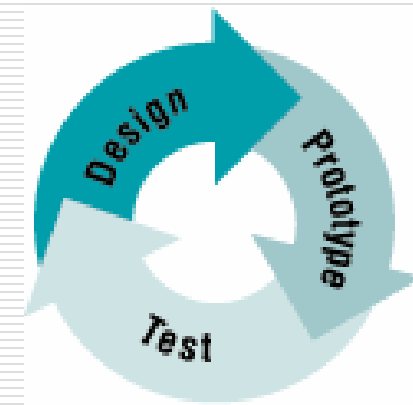


the process of design



steps ...

- set up requirements
 - what is there and what is wanted ...
- analysis
 - ordering and understanding
 - task/activity, scenario/context
- design
 - what to do and how to decide
- prototyping & testing → iteration
 - you'll never get it right the first time
 - finding what is really needed
- implementation and deployment
 - making it and getting it out there



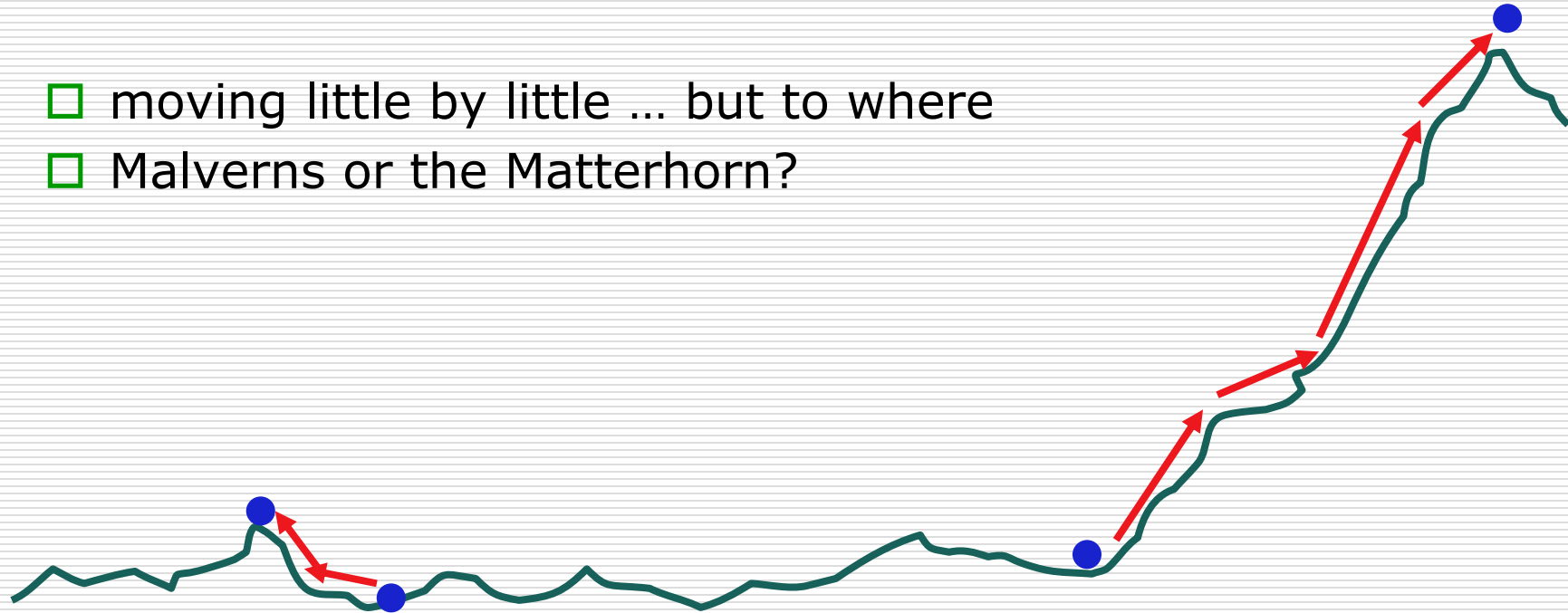
... but how can I do it all ??

- you can't!
 - always limited time
 - trade-off between speed of design and final quality
- can I just find usability problems and fix them?
 - finding problems not hard (there are plenty)
 - issue is: which usability problem is worth fixing?



pitfalls of prototyping

- moving little by little ... but to where
- Malverns or the Matterhorn?



1. need a good start point → good design rules
2. need to understand what is wrong → good evaluation



User-centered design (UCD)

- design philosophy in which the needs, wants and limitations of the end user of a computer product or computer interface are given extensive attention at *each* stage of the design process
- a multi-stage problem solving process that not only requires designers to analyze and foresee how users are likely to use an interface, but to test the validity of their assumptions with regards to real user behavior



Various methods of user-centered design

- Field studies (including contextual inquiry)
- User requirement analysis
- Iterative design
- Usability evaluation
- Task analysis
- Focus groups
- Formal heuristic evaluation
- User interviews
- Surveys
- ...

Ranking based on survey among experienced UCD practitioners
(103 questionnaires) (Mao et al., 2005)



usability engineering

- test of usability based on measurement of user experience
- demands that specific usability measures be made explicit as requirements

usability specification =

- usability attribute/principle
- measuring concept
- measuring method
- now level/worst case/planned level/best case

problems

- usability specification requires level of detail that may not be possible early in design
- satisfying a usability specification does not necessarily satisfy usability



usability specification for a VCR

Usability attribute: Backward recoverability

Measuring concept:	Undo an erroneous programming sequence
Measuring method:	Number of explicit user actions to undo current programming
Now level:	No current product allows such an undo
Worst case:	As many actions as it takes to program in mistake
Planned level:	A maximum of two explicit user actions
Best case:	One explicit cancel action



ISO usability standard 9241

adopts traditional usability categories:

effectiveness

■ can you achieve what you want to?

efficiency

■ can you do it without wasting effort?

satisfaction

■ do you enjoy the process?



some metrics from ISO 9241

Usability objective	Effectiveness measures	Efficiency measures	Satisfaction measures
Suitability for the task	Percentage of goals achieved	Time to complete a task	Rating scale for satisfaction
Appropriate for trained users	Number of power features used	Relative efficiency compared with an expert user	Rating scale for satisfaction with power features
Learnability	Percentage of functions learned	Time to learn criterion	Rating scale for ease of learning
Error tolerance	Percentage of errors corrected successfully	Time spent on correcting errors	Rating scale for error handling



design rules

why rules? need a good start point!

- design for usability...
 - relies on maximizing benefit of one good design by abstracting out properties that direct purposeful design
 - requires both creative insight and principled practice
- design *rules*
 - direction for design
- design *patterns*
 - capture and reuse design knowledge
 - often with no theoretical basis



design patterns

- approach to reuse knowledge about successful design solutions
- Examples
 - Light on Two Sides of Every Room (architecture)
 - Go back to a safe place (HCI)
- Characteristics of patterns
 - capture design practice, *not* theory
 - capture essential common properties of good examples
 - represent design knowledge at varying levels: social, organizational, conceptual, detailed



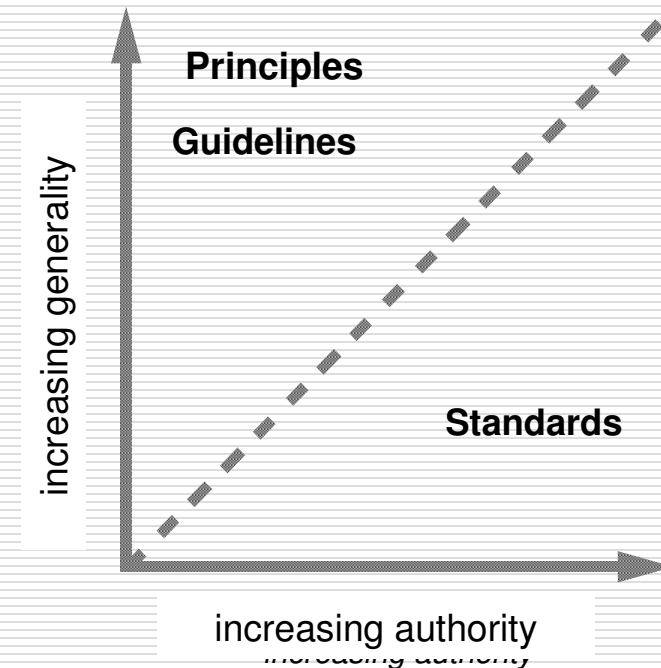
design rules

1. standards

- specific design rules
- high authority
- limited application

2. principles & guidelines

- abstract design rules
- low authority
- high generality



standards

- ❑ set by national or international bodies to ensure compliance by a large community of designers
- ❑ standards require sound underlying theory and slowly changing technology
- ❑ hardware standards more common than software
- ❑ high authority, specific rules, but low level of detail
- ❑ Example: ISO 9241



principles and “golden rules”

- “broad brush” design rules
- useful check list for good design
- better design using these than using nothing!
- different collections e.g.
 - Shneiderman’s 8 Golden Rules
 - Norman’s 7 Principles
 - ...



take account of differences of user types
(*"Know thy user", Hansen, 1971*)

- User profiles
 - Novices
 - Occasional users
 - Expert users
- Use profiles and user demands
 - Frequent use – short cuts/function keys
 - Occasional use – ctrl + key
 - Seldom use – offer function in menu
- Use appropriate interaction styles
 - Form filling, direct manipulation, command entry, etc.



Shneiderman's 8 rules of interface design

- 1. Strive for consistency*
- 2. Enable frequent users to use shortcuts*
- 3. Offer informative feedback*
- 4. Design dialogs to yield closure*
- 5. Offer error prevention and simple error handling*
- 6. Permit easy reversal of actions*
- 7. Support internal locus of control*
- 8. Reduce short-term memory load*



Shneiderman's 8 rules (ctd)

1. Konsistenz anstreben

- Terminologie
- Menüs
- Hilfetexte
- Farbgestaltung
- Layout
- GROßSCHREIBUNG
- Schrifttypen

2. Tastaturkürzel oder andere *shortcuts* für Experten

- Anzahl der Interaktionsschritte verringern
- Sondertasten bei speziellen Anwendungen mit abgestimmter Hardware

3. Gib aussagekräftiges Feedback

4. Gestalte abgeschlossene Benutzerdialoge

- Spezifiziere Anfang, Mitte und Ende
- Dialogabschluß muß deutlich werden (Bsp.: Bestellbestätigung)



Shneiderman's 8 rules (ctd)

5. Fehlervermeidung

- Verwende Kontrollmechanismen für Benutzereingaben
Bsp.: Buchstaben nur in Buchstabenfelder, in Datumsfeld nur eine Zahl im Datumsformat
- Verwende Interaktionsstil, der nur korrekte Eingaben erlaubt
Bsp.: Menüauswahl besser als Formulareingabe bei wenigen, fest definierten Wahlmöglichkeiten

6. Es muß ein Zurück geben

- Für jede Aktion ein *Rückgängig* besser als "Wollen sie wirklich...?"
- beste Lösung: Änderungshistorie, mehrere Rückgängig-Schritte

7. Benutzer sollen Aktionen initiieren, nicht auf das System reagieren (Kontrollverlust)

8. Beachte die Kapazität der geforderten Gedächtnisstrukturen



prevent errors from happening

□ demand complete input

- Automatic recognition of invalid pair structures, e.g., brackets must be closed
- Allow grouping of command sequences as one action
- Only allow commands that are executable, provide information and help
e.g.: automatic input completion (tab expansion), display of possible actions (reverse thrust in aircraft)

□ if error messages are necessary...

- Formulate clear and meaningful, point to cause of problem (*"syntax error" vs. "unmatched parentheses"*)
- Don't use negative, repelling or insulting expressions (*"illegal entry", "bad command name"*)



Norman's 7 principles of design

- 1. Use both knowledge in the world and knowledge in the head.*
- 2. Simplify the structure of tasks.*
- 3. Make things consistent and visible: bridge the gulfs of Execution and Evaluation.*
- 4. Get the mappings right.*
- 5. Exploit the power of constraints, both natural and artificial.*
- 6. Design for error.*
- 7. When all else fails, standardize.*



use *metaphors*

- ❑ relate computing to other real-world activity
- ❑ evoke a *mental model* of the system's structure and operations
- ❑ must be consistent!
- ❑ relate to user's experiences
- ❑ facilitate learning and retention of interface



The Desktop Metaphor

L.Allison 1995



important interaction metaphors

- *Desktop metaphor*: currently predominant
- *Book metaphor*: for big documents, e.g. hypertext
- *Filing cabinets*: for online documentation, system settings, etc.
- *Office metaphor*: for collections of documents
- *Library metaphor*: for large collections of documents
- *Building metaphors, etc.*: for virtual worlds
- *Animated agent metaphor*: for guidance and recommendation
- *Composite metaphors*: combine metaphors, e.g. office + file cabinet + desktop



use *affordances*

- "term that refers to the properties of objects -- what sorts of operations and manipulations can be done to a particular object" (D. A. Norman 1988, *The Psychology of everyday things*)
- A „door“ affords opening, a „chair“ affords support
- *perceived affordance*: the extent to which objects visually suggest their affordance



mug handle



'affords'
grasping



perceived affordances

- for physical objects
 - shape and size suggest actions (pick up, twist, throw...)
- for screen objects
 - button-like object 'affords' mouse click
 - physical-like objects suggest use
- culture of computer use
 - icons 'afford' clicking
 - or even double clicking ... not like real buttons!

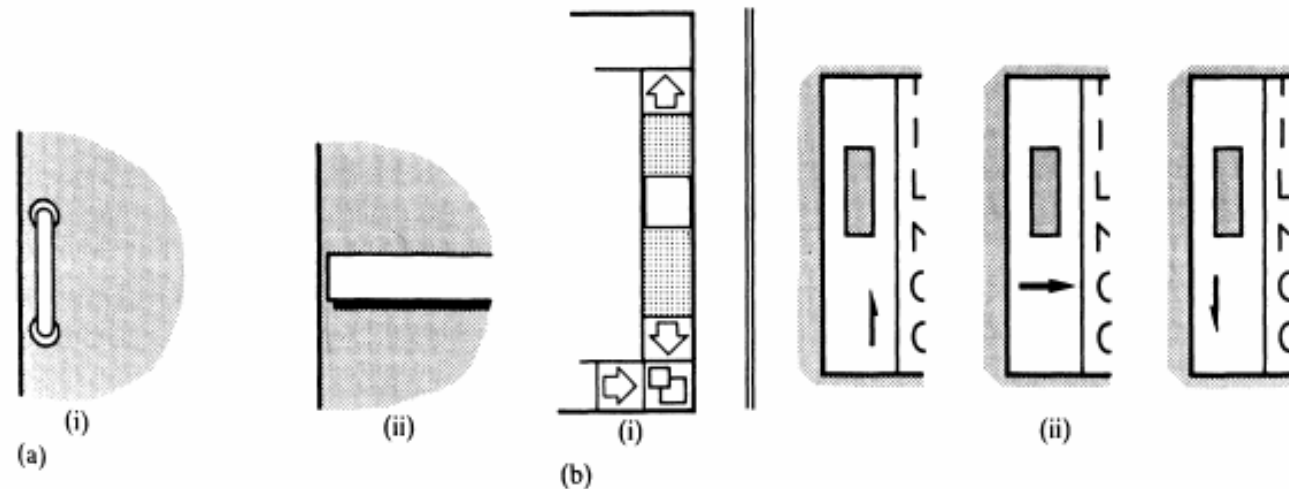
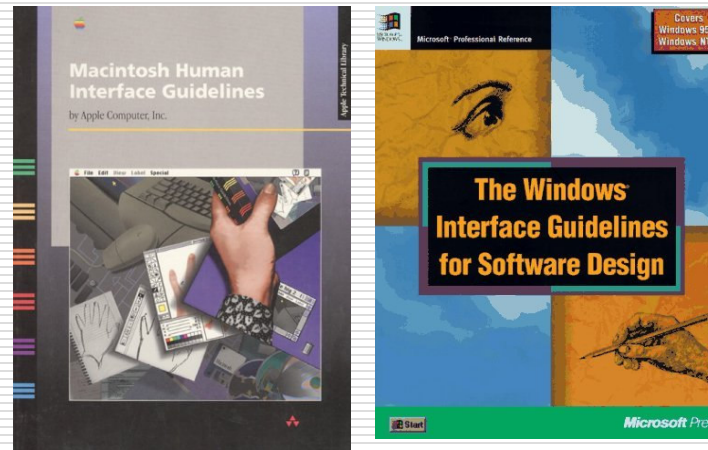


Figure 4.5 The affordance of objects (Gaver, 1991). (a) Door handles; (b) scroll bars (© 1991, Association for Computing Machinery, Inc. reprinted by permission).



guidelines

- ❑ more suggestive (e.g. for following one metaphor)
- ❑ many textbooks and reports full of them
- ❑ abstract guidelines applicable during early life cycle activities
- ❑ detailed guidelines (style guides) applicable during later life cycle activities
- ❑ understanding justification for guidelines aids in resolving conflicts



Eingabe von Informationen (Smith & Mosier, 1986)

1. *Konsistenz*

Bsp.: ähnliche Vorgehensweise bei allen Dateneingaben; nicht für Formulareingabe einmal Mausfokus notwendig, einmal nicht

2. *Minimierung der Eingabeaktionen*

Bsp.: keine Redundanzen; angemessene Eingabewidgets

3. *Minimale Gedächtnisbelastung für den Benutzer*

Bsp.: keine komplexe Eingabesyntax und schwer memorierbare Eingabecodes

4. *Kompatibilität mit Ausgabe*

Bsp.: s.o.

5. *Flexibilität*

Bsp.: Eingabesequenz und -art für Benutzer modifizierbar; Reihenfolge der Formularfelder bei Druck auf <TAB>



Darstellung von Information (Smith & Mosier, 1986)

1. *Konsistenz*

Bsp.: Terminologie, Abkürzungen, Farben, etc. mit festen Bedeutungen verbinden ("*keine Farbänderung ohne Grund*")

2. *vereinfachte Informationsaufnahme durch "Ordnung"*

Bsp.: Bündigkeit von Text und Zahlen, verständliche Labels

3. *minimale Gedächtnisbelastung für den Benutzer*

Bsp.: alles wichtige an einem Ort; keine Information von einer Bildschirmseite auf einer anderen nötig

4. *Kompatibilität mit Eingabe*

Bsp.: wenn möglich, Ausgabefelder auch editierbare Eingabefelder

5. *Flexibilität*

Bsp.: Der Benutzer sollte die Ausgabe seinen Wünschen anpassen können



Wie errege ich Aufmerksamkeit?

(Wickens & Hollands, 2000)

- max. zwei Stufen
- Markierung: unterstreichen, einrahmen, Index verwenden
- Größe: bis zu vier unterschiedliche Größenstufen
- Fonts: bis zu drei unterschiedliche Schriftarten
- invertierte Darstellung
- Blinken: 2-4 Hz, begrenzt einsetzen und nur in bestimmten Bildschirmbereichen
- Farbe: bis zu vier Farben
- Audio: weiche Töne für positives Feedback, harte seltener für wirklich wichtige Notfälle



Guidelines for screen design and layout

□ Principles

- ask: what is the user doing?
- think: what information, comparisons, order
- design: *form follows function*

□ Tools

- grouping of items
- order of items
- decoration - fonts, boxes etc.
- alignment of items
- white space between items



grouping and structure

logically together \Rightarrow physically together

Billing details:

Name

Address: ...

Credit card no

Delivery details:

Name

Address: ...

Delivery time

Order details:

item

quantity cost/item cost

size 10 screws (boxes)

7 3.71 25.97

.....

...

...

...

order of groups and items

- think - what is natural order

- should match screen order

- use boxes, space etc.
- set up tabbing right!

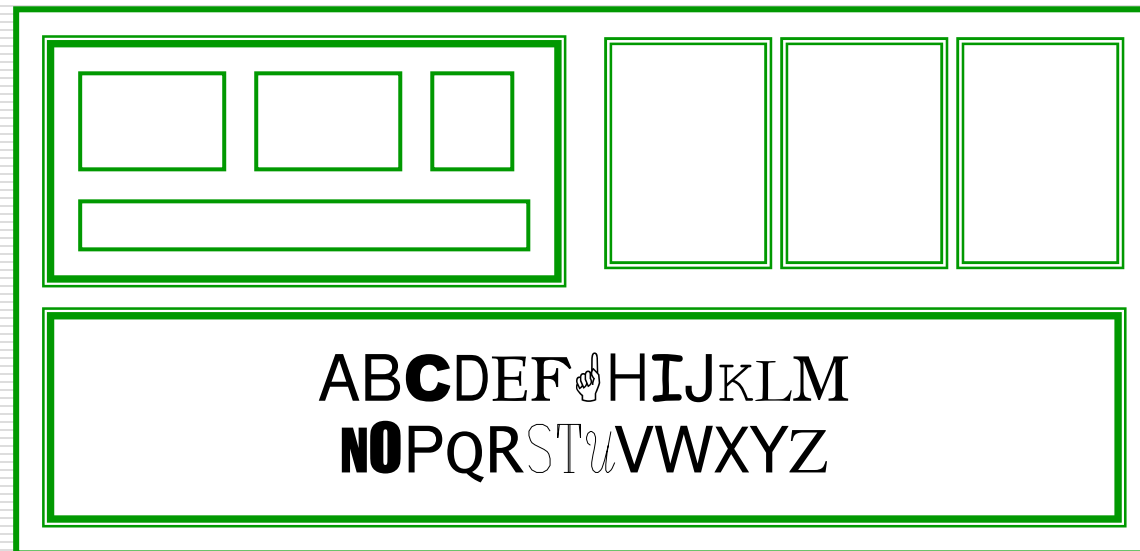
- instructions

- beware the cake recipe syndrome!
... mix milk and flour, add the fruit
after beating them



decoration

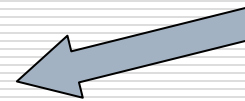
- use boxes to group logical items
- use fonts for emphasis, headings
- but not too many!!



alignment - text

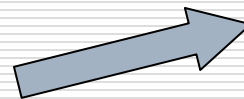
- you read from left to right (English and European)
⇒ align left hand side

Willy Wonka and the Chocolate Factory
Winston Churchill - A Biography
Wizard of Oz
Xena - Warrior Princess



boring but
readable!

fine for special effects
but hard to scan




Willy Wonka and the Chocolate Factory
Winston Churchill - A Biography
Wizard of Oz
Xena - Warrior Princess



alignment - names

- usually scanning for surnames ⇒ make it easy!


Alan Dix
Janet Finlay
Gregory Abowd
Russell Beale



Alan Dix
Janet Finlay
Gregory Abowd
Russell Beale



Dix , Alan
Finlay, Janet
Abowd, Gregory
Beale, Russell



multiple columns

- scanning across gaps hard
- use leaders

sherbert.....	75
toffee.....	120
chocolate.....	35
fruit gums.....	27
coconut dreams.....	85

- or greying (vertical too)

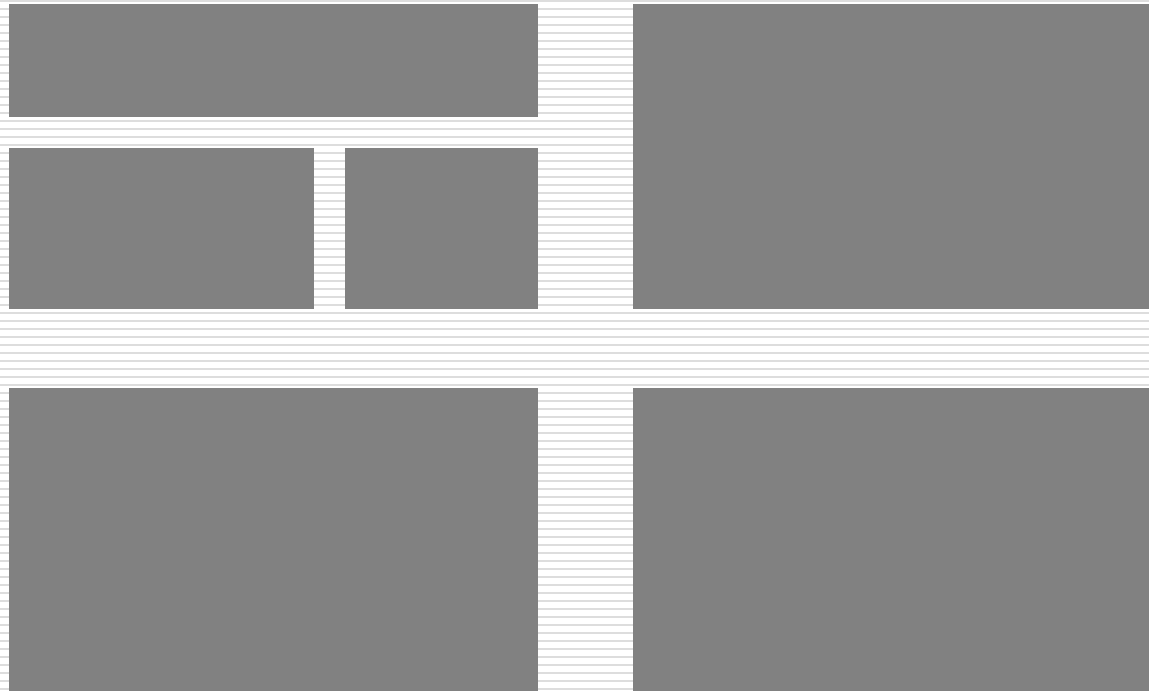
sherbert	75
toffee	120
chocolate	35
fruit gums	27
coconut dreams	85



space to separate



space to structure



space to highlight



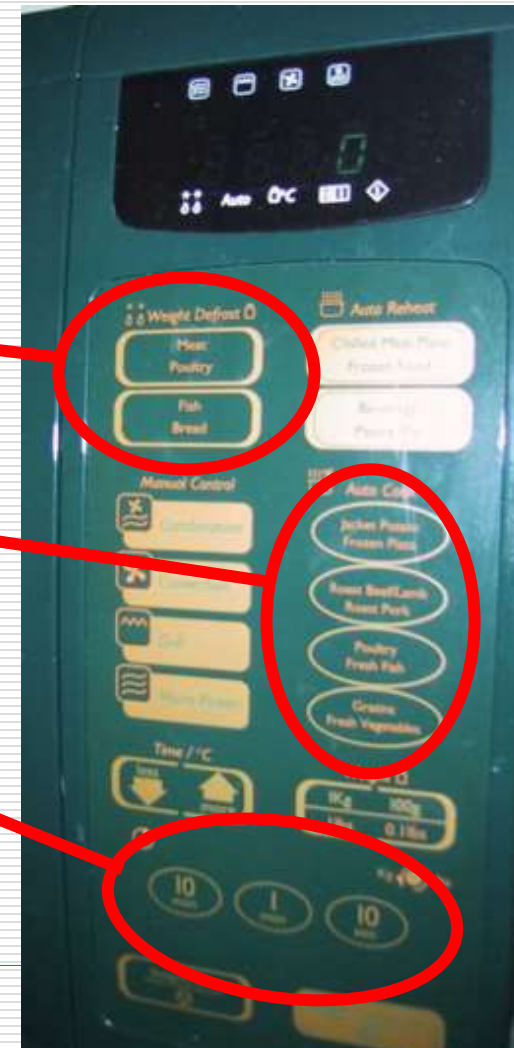
example: physical controls

□ grouping of items

defrost settings

type of food

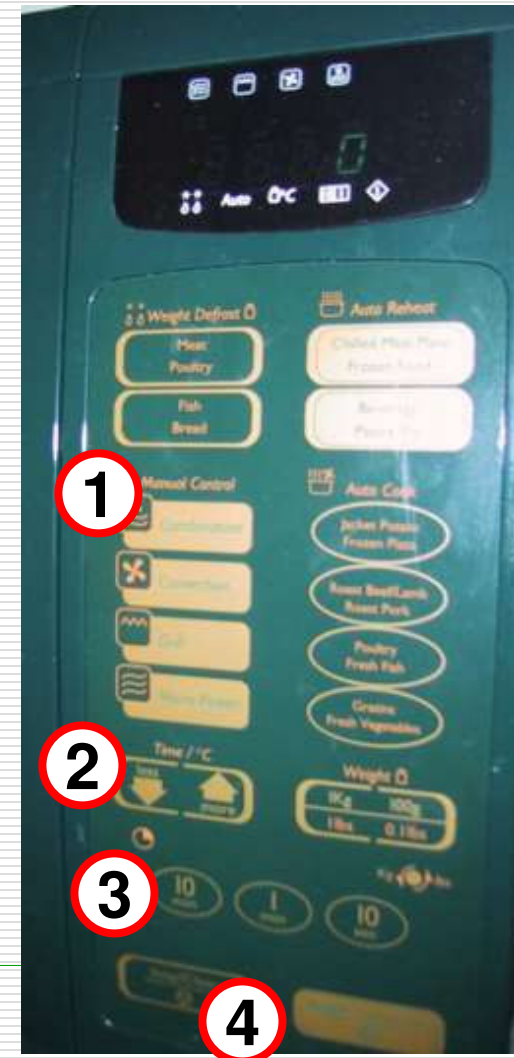
time to cook



physical controls

- grouping of items
- order of items

- 1) type of heating
- 2) temperature
- 3) time to cook
- 4) start

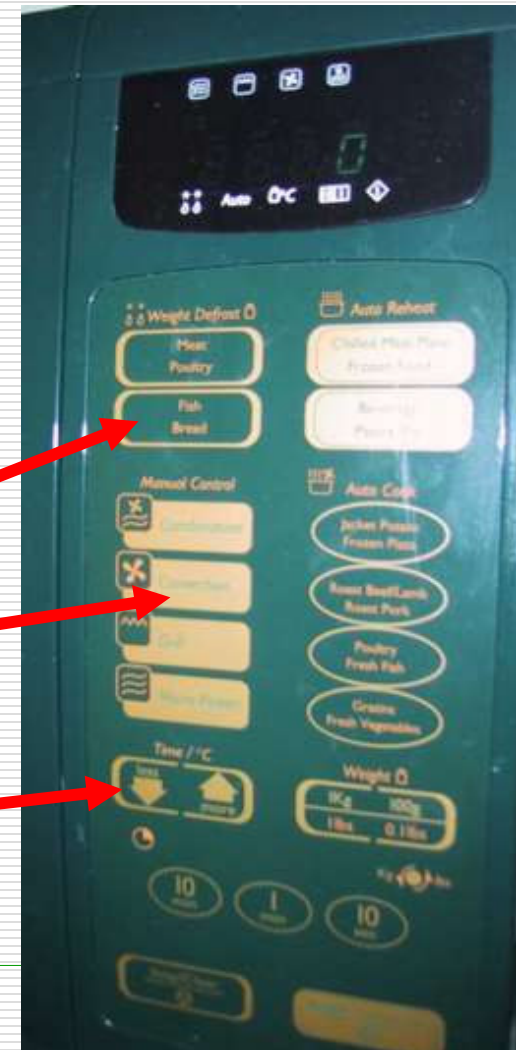


physical controls

- grouping of items
- order of items
- decoration

different colors for
different functions

lines around related
buttons (temp up/down)

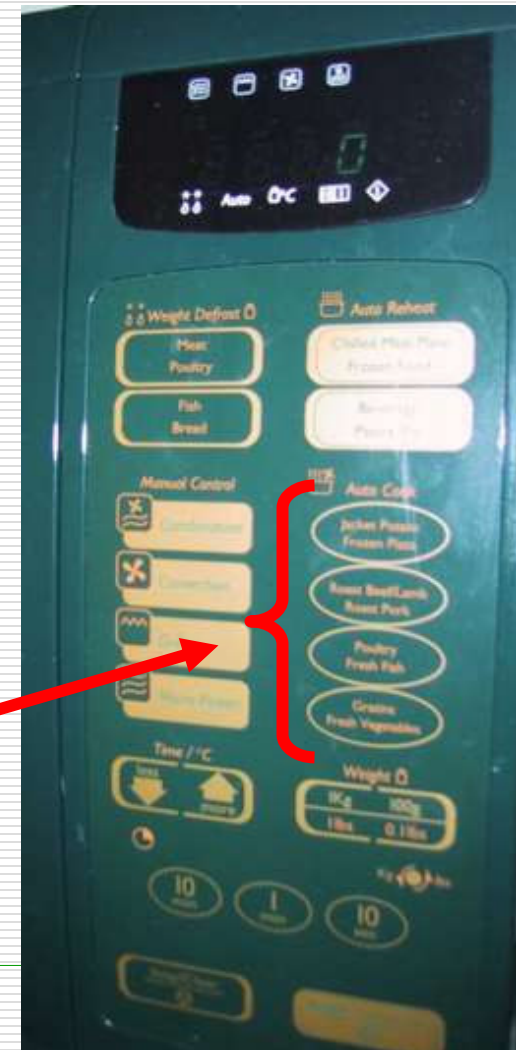


physical controls

- grouping of items
- order of items
- decoration
- alignment

centred text in buttons

? easy to scan ?



physical controls

- grouping of items
- order of items
- decoration
- alignment
- white space

gaps to aid grouping



Ok, that's for how to create a good(?) *first* system

Next...

How to evaluate and improve??

