Digital Interaction Design IS-104

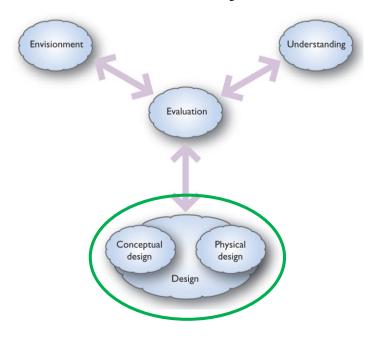
DESIGN PRINCIPLES FOR USABILITY

MOHAMMAD TAFIQUR RAHMAN

OCTOBER 9 2019



Goals for today



Understanding design principles and modelling the system: (Chapter 5 & 9)

- Elaborate on Benyon's 12 design principles (chapter 5)
- Designing user experiences (chapter 9)
 - UX Design (Conceptual)
 - UI Design (Physical)



Learnability

- 1. Visibility
- 2. Consistency
- 3. Familiarity
- 4. Affordance

Effectiveness

- 5.Navigation
- 6.Control
- 7.Feedback
- 8.Recovery
- 9.Constraints

Accommodation

- 10.Flexibility
- 11.Style
- 12.Conviviality

A set of 12 principles grouped in 3 broader categories.



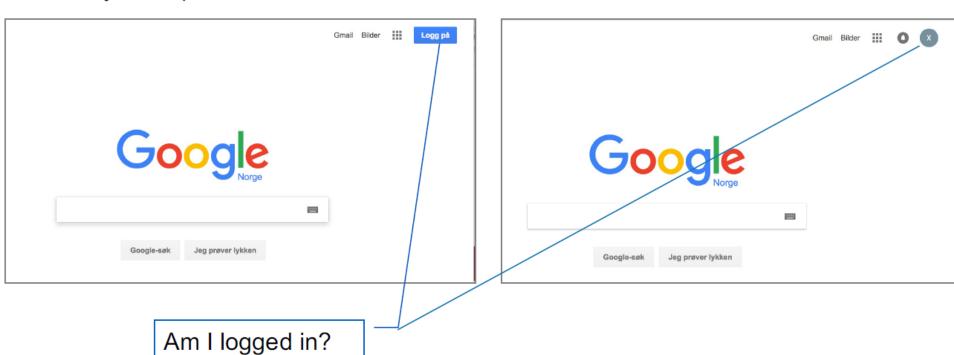
Principles related to learnability:

- **Visibility** Try to ensure that things are visible so that people can see what functions are available and what the system is doing.
- **Consistency** Be consistent in the use of design features and be consistent with similar systems and standard ways of working.
- **Familiarity** Use language and symbols that the intended audience will be familiar with.
- **Affordance** Design things so it is clear what they are for; for example, make buttons look like buttons so people will press them.



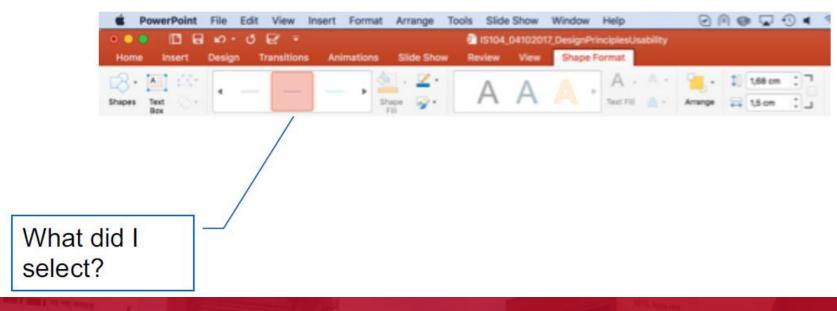


Visibility example



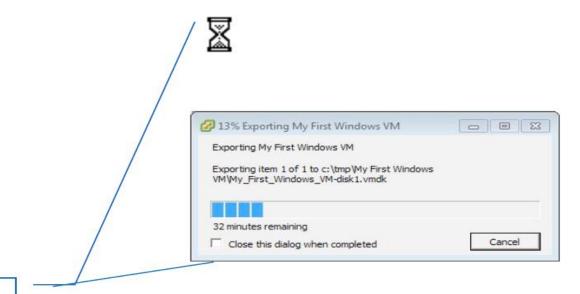


Visibility example





Benyon's Principles Visibility example



What is the system doing?



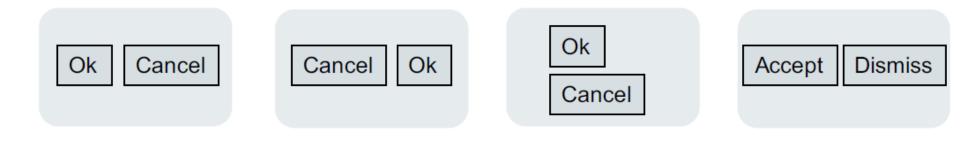
Benyon's Principles Consistency example

- Consistent sequence of actions for similar situations
- Identical Terminology (prompts, menus, help)
- Consistent visual layout (fonts, color, etc.)





Inconsistency example





Benyon's Principles Familiarity example

Follow real world conventions.







Familiarity example

Speak the users' language.

Maximum withdrawal of \$50 at this time

X.25 connection discarded due to network congestion.

Local limits now in effect



Affordance example

· Buttons should look like buttons.

Submit

Submit

Submit

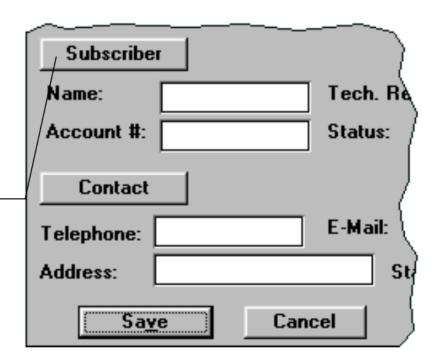


Affordance example

 But "non-buttons" should not look like buttons.

These are labels with a raised appearance.

Is it a surprise that people click on them?





Principles related to effectiveness:

Three effectiveness principles are about ease of use.

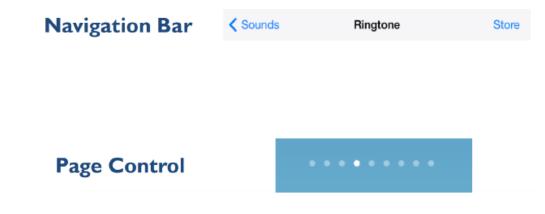
- 5 **Navigation** Provide support to enable people to move around the parts of the system.
- 6 **Control** Make clear what can be controlled and allow people to take control.
- 7 **Feedback** Rapidly feed back information from the system to people so that they know what effect their actions had.





Navigation examples

 Users should always know where they are.





Navigation examples

 Users should be able to move around by making selections.





Control examples

Many interfaces interrupt users with questions. Many of these dialogs offer the option "Never ask me again".







Control examples

Many systems perform automatic operations that might interrupt users. It is important to allow users to control such interruptions.

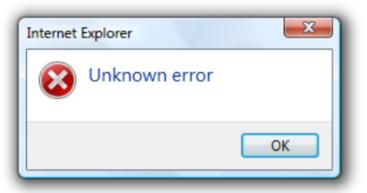




Feedback examples

Error messages are good examples of feedback. They need to:

- Be written in plain language
- · Indicate the problem
- Suggest a solution





Feedback examples

| Email address: (You will use this email to sign in.) | |
|--|-------------------|
| ⚠ This email address is invalid. Please try a different one. | |
| Password: | Confirm password: |
| (Between 6 and 15 characters with no | |
| spaces.) | |
| ▲ Please enter a valid password. | |
| Member name: (Displayed to other members, not unique.) | |
| ⚠ Please enter a display name. | |



Principles related to effectiveness:

Two effectiveness principles are about safety.

- 8 **Recovery** Enable recovery from actions, particularly mistakes and errors, quickly and effectively.
- 9 **Constraints** Provide constraints so that people do not try to do things that are inappropriate. In particular, people should be prevented from making serious errors through properly constraining allowable actions and seeking confirmation of dangerous operations.



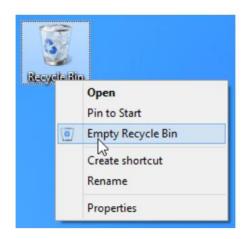
Oh no!



Recovery examples

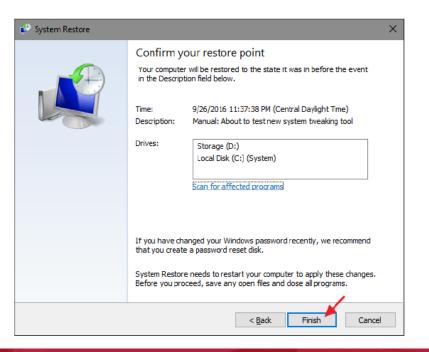
 As much as possible, actions should be reversible.







Recovery example





Benyon's Principles Constraints examples

Prevent errors:

- Try to make errors impossible.
 E.g. can only enter legal data.
- Gray out menu items that don't apply.







Principles related to accomodation:

10 **Flexibility** – Allow multiple ways of doing things so as to accommodate people with different levels of experience and interest in the system. Provide people with the opportunity to change the way things look or behave so that they can personalize the system.

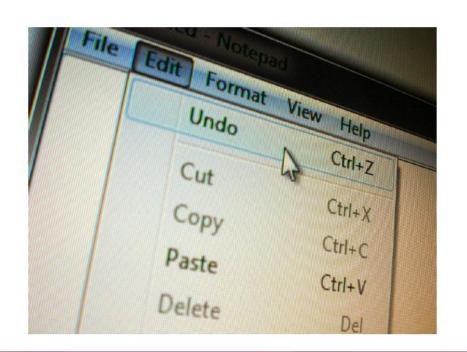
- 11 **Style** Designs should be stylish and attractive.
- 12 **Conviviality** Interactive systems should be polite, friendly, and generally pleasant. Design for politeness.





Flexibility examples

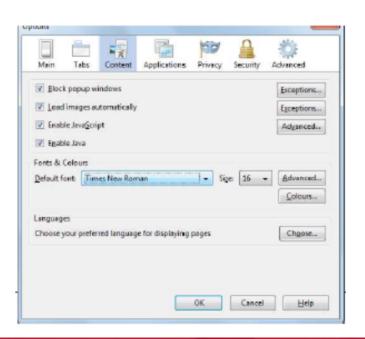
Accelerators for experts e.g., keyboard shortcuts





Flexibility examples

Give to users the ability to modify the user interface.





Style examples

Style relates to design trends





Benyon's Principles Style examples



2009







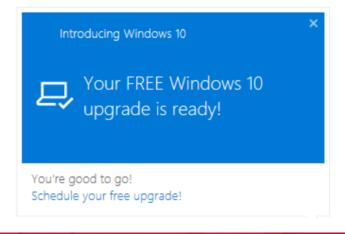
Conviviality examples

Nothing ruins the experience of using an interactive system more than an aggressive message or an abrupt interruption.





Conviviality: example of impolite software



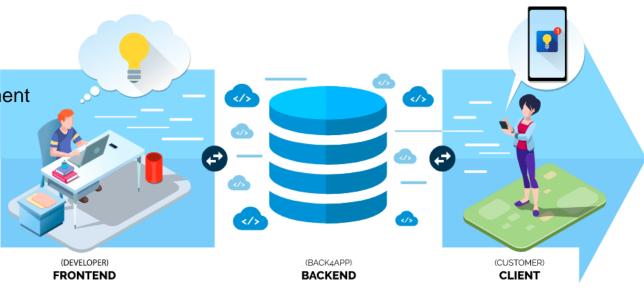
nagware

presents the user with one or more pop-up windows or alerts when an application is launched or closed (sometimes both), reminding the user to register, purchase, or take some other action which is not related to the task at hand.



Software Systems Development

- UX design
 - Research
 - Analysis
 - Concept development
- Frontend design
 - UI design
 - Programmability
 - Functionality
- Backend design
 - Database design
 - Data management
 - Data storage





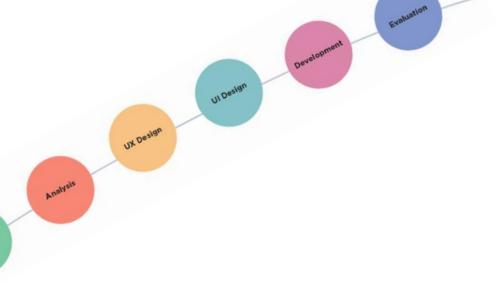
Designing User Experiences

- UX Design
- Metaphors
- Conceptual design
- Physical design
- Designing interactions





- UX Design
 - Exploring design space
 - Exploring design concept
 - Places for experiences



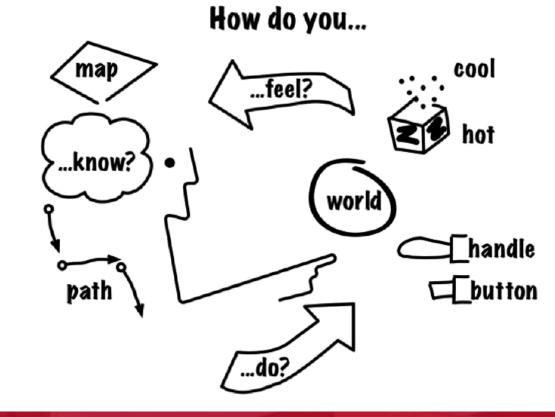


- Exploring design space
 - design dimensions
 - exploring alternatives
 - design fixation
 - open to radical solutions: service blueprints, customer journey mappings, wireframes and navigation maps
 - brainstorming, envisionment, sketching, information architecture





- Exploring design space
 - How do you do?
 - How do you feel
 - How do you know





- Places for experiences
 - Physical space
 - Digital space
 - Information space
 - Social space





Design Metaphors

- What is it?
 - "The way we think, what we experience and what we do every day"
 - Taking concepts from one domain (source) and applying them to another (target)
 - Use metaphor to describe new domain in terms of familiar ones
 - Not just a simple mapping from one domain to another: computer window & window in a house





Design Metaphors

- Metaphors help users to
 - understand abstract content,
 - create a sense of familiarity
 - trigger emotions & draw attention
 - motivate actions
- Dos: make the unfamiliar familiar, awake positive associations and persuade people
- Don'ts: simplistically literal metaphor and blindly mimic a real-world precedent,



A ninja stole this page.

Conceptual Design

- Conceptually model the investigating domain
 - Explore, generate and document ideas
 - Concentrates user needs, the objectives of the service or system, requirements for content and functions, information architecture and interaction design
 - Help or hinder users in developing their own mental model of the system or service





Conceptual Design

Scenarios and conceptual design

Individual version of scenario

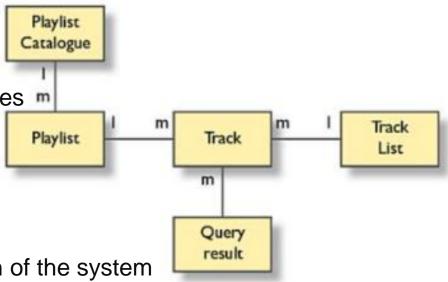
Covers major uses and functionalities m

Object-action analysis

Object: noun/noun phrases

Actions/activities: verbs

Diagrammatic technique: representation of the system





Physical Design

Components of physical design

Operational design

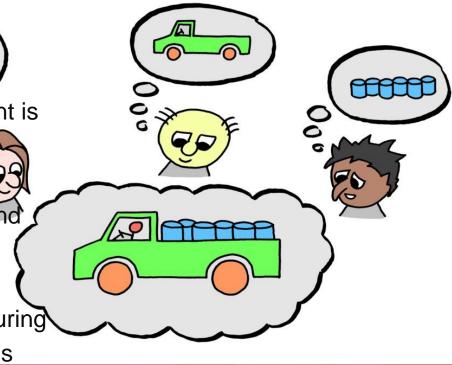
How everything works and sontent is structured and stored

Representational design

Fixing on colors, shapes, sizes and information layout

Interaction design

 Allocation of functions and structuring and sequencing of the interactions





Physical Design

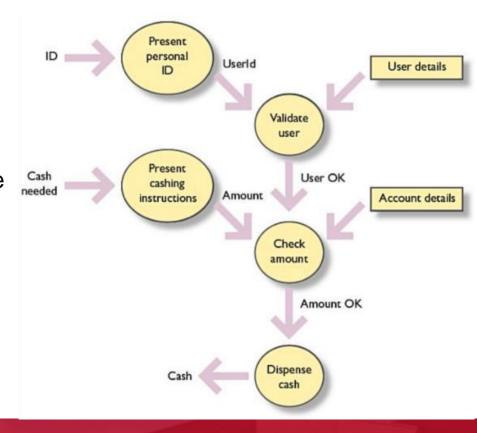
- Design Language
 - Building meanings into objects
 - What things to do
 - Distinctions between types of objects
 - Consists of:
 - design elements
 - principles of composition
 - qualifying situations





Designing interactions

- Allocate functions and knowledge to persons or devices
- Consider capabilities of people and the constraints on what they can do
- The interaction should be engaging, enjoyable and fulfilling
- Interaction patterns are built up with complex interactions of menus and mice/multi-touch display.
- Diagrammatic techniques: representing the process (dataflow diagram or sequence diagram or state transition network)





Tips for UX Design

- Usability: purposeful and functional
- User profiling: who are users & what do they want to achieve
- Asking for permission: only when it is needed
- Form vs function: always go for function over form
- Consistency: inconsistency in design creates confusion and unhappy users
- Simplicity: app is grandma-proof
- Don't make me think: beautiful, clear and easily understandable interface and functions

(source: https://www.freecodecamp.org/news/whats-the-difference-between-ux-and-ui-design-2ca8d107de14/)





