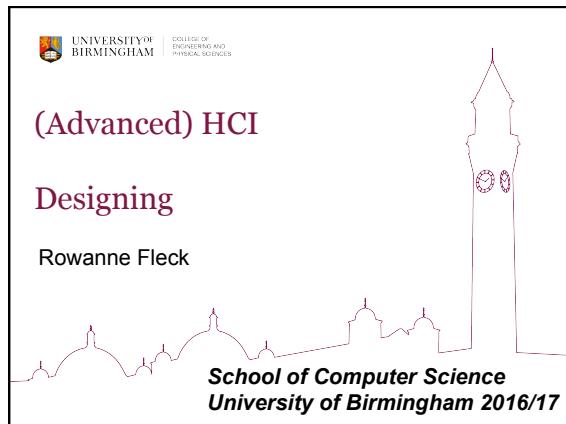


UNIVERSITY OF BIRMINGHAM | COLLEGE OF ENGINEERING AND PHYSICAL SCIENCES

(Advanced) HCI

Designing

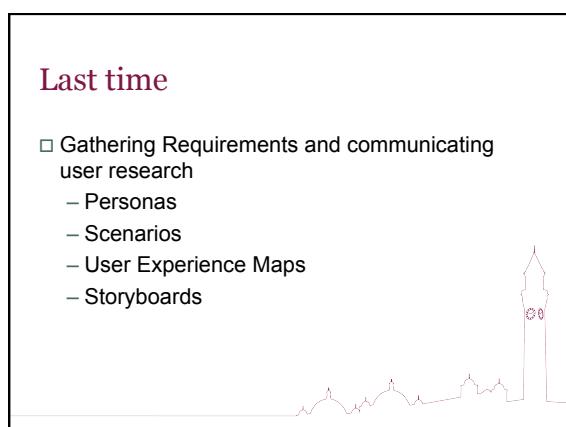
Rowanne Fleck

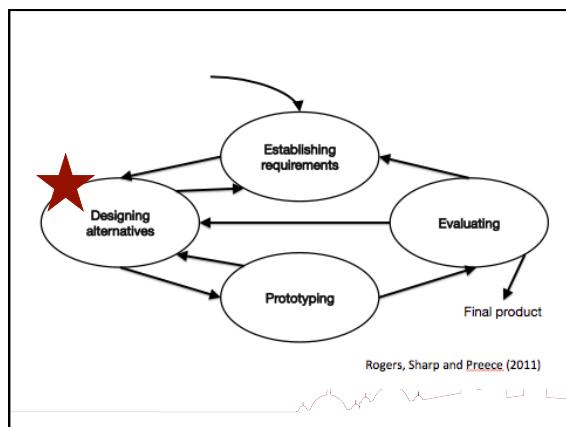


**School of Computer Science
University of Birmingham 2016/17**

Last time

- Gathering Requirements and communicating user research
 - Personas
 - Scenarios
 - User Experience Maps
 - Storyboards





How do you *design* something?



Designing alternatives

- Process of generating ideas to meet requirements
- Conceptual design
 - What people can do with product
 - What concepts needing to be known to interact with it
- Physical Design
 - Colours, sounds, image, menu and icon design



How do we come up with designs?

- Design flair...but not always
- Innovations also come from:
 - Seeking different perspectives
 - Evolution of existing products
 - Looking at other designs for inspiration
- Much previous knowledge captured in
 - Design guidelines and principles
- Need to remember our users and their requirements



Guidelines and principles



Design Principles

Intended to help designers explain and improve design (Thimbleby, 1990)

1. Visibility
2. Feedback
3. Constraints
4. Consistency
5. Affordance



Visibility

Visibility important cue to identify what to use and how



Good Visibility



Bad Visibility

Feedback

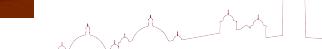
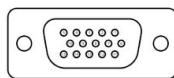
Confirmation of action and state of system

Can be auditory, verbal, visual or combination



Constraints

Forced Function (Norman, 1998)



Consistency

Similar operations within & across interactions

Consistent function is easier to learn and use

Difficult with interfaces with 100's operations



Affordance

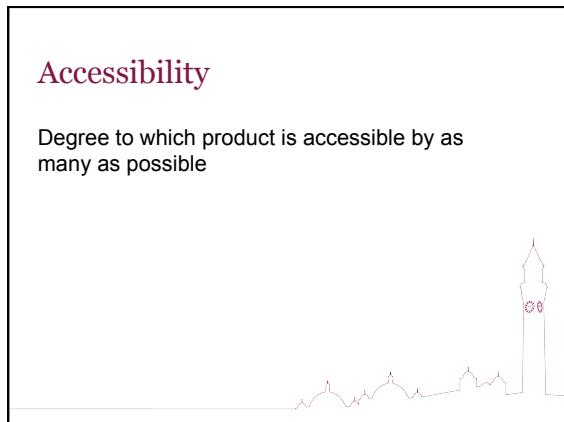
Functional inference through attributes
Physical (a) or Perceived (b)





Accessibility

Degree to which product is accessible by as many as possible



Standards

Set by national or international bodies

Hardware standards more common than software

ISO 9241 defines usability as: **effectiveness, efficiency and satisfaction with which specified users can achieve specified goals in a particular environment.**



Golden rules and heuristics

“Broad brush” design rules

Useful check list for good design

Different collections e.g.

- Nielsen's 10 Heuristics (more for evaluation)
 - Shneiderman's 8 Golden Rules
 - Norman's 7 Principles



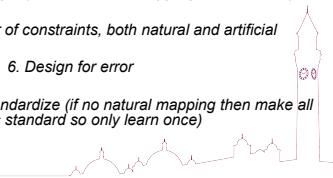
Shneiderman's 8 Golden Rules

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



Norman's 7 Principles

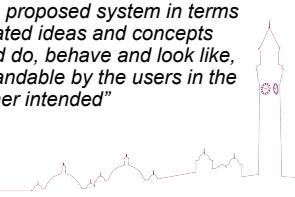
1. Use both knowledge in the world and knowledge in the head
2. Simplify the structure of tasks
3. Make things visible: bridge the gulfs of Execution and Evaluation
4. Get the mappings right (user intentions mapping onto controls)
5. Exploit the power of constraints, both natural and artificial
6. Design for error
7. When all else fails, standardize (if no natural mapping then make all things standard so only learn once)



Conceptual Design

Transform **user requirements** into a conceptual model

"a description of the proposed system in terms of a set of integrated ideas and concepts about what it should do, behave and look like, that will be understandable by the users in the manner intended"



Interface Metaphors

They combine:

- familiar knowledge
- new knowledge

To help the user understand the product and how to use it



Evaluate the Metaphor

What structure does it provide?

– A good one should give familiar model

Relevant to the problem?

Easy to represent?

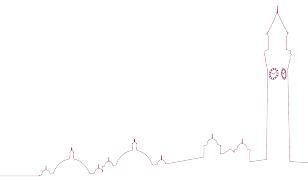
– Audio, visuals and words

How extensible is it?

– Useful for later in the interaction?



Will the audience understand it?





Conceptual Models

Which interaction type?

- instructing
 - conversing
 - manipulating
 - exploring

Most models have combination of these

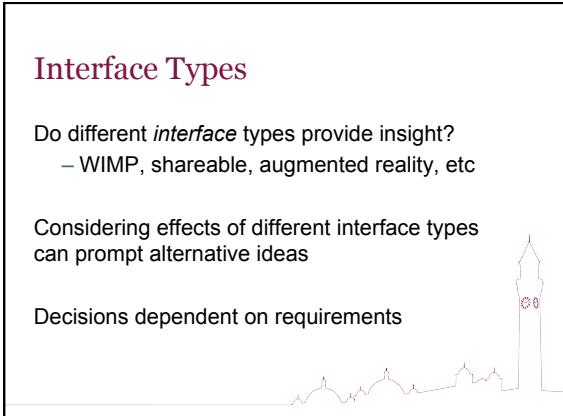


Interface Types

- Do different *interface* types provide insight?
 - WIMP, shareable, augmented reality, etc

Considering effects of different interface types can prompt alternative ideas

Decisions dependent on requirements

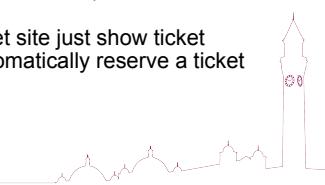




Expanding the model

What functions will the product perform?

- What will the product do and what will the human do (task allocation)?
- E.g. should ticket site just show ticket options? Or automatically reserve a ticket too?



Expanding the model

How are the functions related?

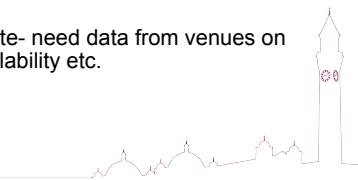
- Sequential/temporal or parallel
- Categorisations, e.g. all actions related to telephone memory storage



Expanding the model

What information needs to be available?

- What data is required to perform the task?
 - How is this data to be transformed by the system?
 - E.g. travel site- need data from venues on access, availability etc.

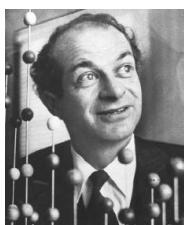


Conceptual to Physical Design

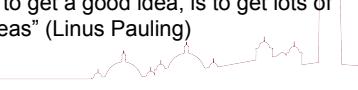
Conceptual design should develop freely

Tying physical design too early might inhibit creativity

Designing alternatives



“....the best way to get a good idea, is to get lots of ideas” (Linus Pauling)



The Hill Climbing of Iterative Design

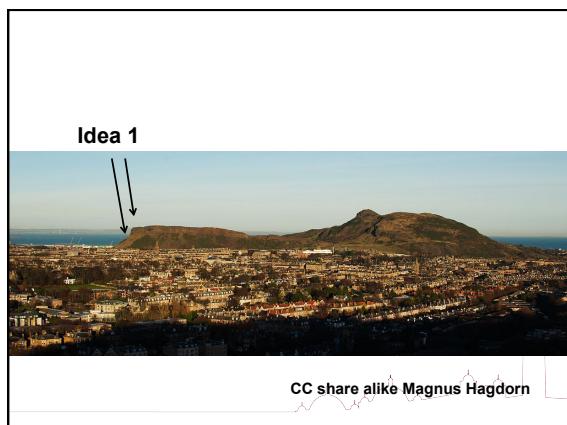
- Start somewhere
- Evaluate designs to make better
- Keep on doing it until you can't get any better
- Sounds ok?.....



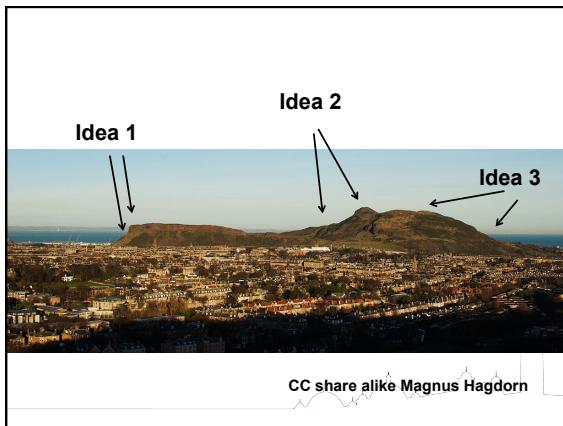
Copyright Rob Farrow and licensed for reuse under this Creative Commons Licence.



<http://www.geograph.org.uk/photo/2790395>



CC share alike Magnus Hagdorn



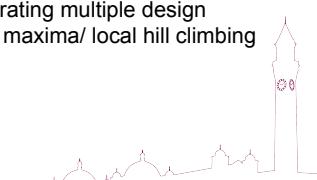
Getting the design right vs. getting the right design

- getting the right design involves **searching as much of a design space as possible**
- once you've found a promising design you want to improve it as much as possible (get it right) by **exploring variations** (local hill climbing)



Summary

- Talked about how to design
 - Design principles and guidelines
 - Conceptual design and use of metaphors
- Importance of generating multiple design ideas to avoid local maxima/ local hill climbing



Next time...

- Understanding human constraints and cognition and design implications – Andrew Howes

