Lecture 3: Goal-oriented interaction

Using cognitive theories of planning, learning and understanding to understand user behaviour, and what they find hard.
Overview of the course

• Theory driven approaches to HCI
• Design of visual displays
• **Goal-oriented interaction**
• Designing efficient systems
• Designing smart systems (guest lecturer)
• Designing meaningful systems (guest lecturer)
• Evaluating interactive system designs
• Designing complex systems
A *Metatheory* (in first-wave HCI): User interaction can be modelled as search
Reminder from Prolog course: problem solving using graph search
Turn the problem into a graph
Encode as Prolog facts to solve

route(a,g).
route(g,l).
route(l,s).
... 
travel(A,A).
travel(A,C) :- route(A,B),travel(B,C).

solve :- start(A),finish(B), travel(A,B).
HCI example of a **User Goal**: “How much did my use of Google Cloud Platform cost me last month?”
Build What's Next
Better software. Faster.

- Use Google's core infrastructure, data analytics and machine learning.
- Secure and fully featured for all enterprises.
- Committed to open source and industry leading price-performance.

GO TO CONSOLE
CONTACT SALES

Forrester Research
Google Cloud is named the Insight PaaS Leader by Forrester.
LEARN MORE

GCP Region Expansion
Run workloads in even more locations around the world. Our newest regions: Frankfurt, São Paulo and Mumbai.
LEARN MORE

Let's Talk About AI
Join the Cloud OnAir: The Journey From Big Data to AI global event on December 5.
LEARN MORE

Why Google Cloud Platform?
What search algorithm is being used here?
Breadth first/Depth first?
Click targets
Click targets
[Simplified] Cognitive Walkthrough

Goal

Availability

Match

Feedback

See: https://www.colorado.edu/ics/sites/default/files/attached-files/93-07.pdf

For a detailed description
Finding your bill?

Goal

Availability

Match

Feedback
Example: Walkthrough of an API (demo)

Goal

Availability

Match

Feedback

(Macvean et al, 2016)
Example problem: Discovery

**Goal**
I want to delete a file

**Availability**
Type “File.” and auto complete gives

```java
void main() {
    File.
    fromRawPath()
    fromUri()
}
```

**Match**
There’s a conceptual mismatch on whether file is a static method or you have to get a file and then delete it

**Feedback**
Example problem: ‘yak shaving’

Goal
- To write a line to a file
  - Open a file
    - Complete a future to get the file
      - Convert a string to a bytebuffer
        - Iterate over the bytebuffer
          - Write the block
            - Complete on the future for writing
              - Close the file
                - Complete the future for closing the file

Availability

Match

Feedback

Too many subgoals that need completing

(Macvean et al, 2016)
Example (not-examinable)

(The Factory Pattern in API Design: A Usability Evaluation, Ellis et al)
The cost of thinking: Heuristics and Biases
12 + 24 \times 3 = 84
= AC + 2

“eh?”

(Example from Richard Young)
How many times should the calculator user press AC?
Classical theories of metareasoning

- **Optimal search**
  - Find the best possible solution within stated constraints on resources
- **Bounded rationality**
  - Computation is one of the constraints
- **Satisficing**
  - Find a satisfactory solution within computation constraints
Neuro-economic models of reasoning

• Behavioural economics, popularly known as “Nudge”
• Original basis in “prospect theory” (Kahneman & Tversky)
  • General theory of decision making
  • Construct a utility model, based on outcome of possible actions
  • Weight estimated values by likelihood
  • Choose action with optimal utility
  • May include future value discounting
• In practice, the optimisation is more likely to involve satisficing, due to reasoning with bounded rationality constraints
  • In Kahneman’s terms “thinking fast and slow”
Bounded rationality in humans

- Apply *heuristics* rather than searching for optimal plan
  - Availability heuristic - reason based on examples easily to hand
  - Affect heuristic – base decision on emotion rather than calculating cost / benefit
  - Representativeness heuristic - judge probability based on resemblance
- Apply *biases* to ensure estimation error within tolerable bounds
  - Loss aversion - losses hurt more than gains feel good
  - Expectation bias - researchers (even in HCI) find results they expected
  - Bandwagon effect - do what other people do
- And many others!
Behavioural economics in programming

• “Attention Investment theory” of abstraction use
  • Automation requires abstract specification
    • e.g. defining a regular expression for search and replace
  • Benefit of automation is saving time and concentration in future
  • But abstract specification (programming) takes time and concentration!
    • And powerful abstractions (programs) can go wrong powerfully
  • User may prefer repetitive manual operations - safe and incremental
• So utility function will compare future saving of attention from programming vs costs of concentrating on a risky strategy
  • Biases such as loss aversion will apply
  • Bounded rationality will apply, since deciding what to do takes even more concentration
The limitations of goal based HCI
It assumes the user doesn’t make mistakes

- Would need a cognitive model of why error occurred
  - Information loss due to cognitive limitations
  - Incorrect mental model
  - Misleading design
- Need description of user journey that accounts for problem identification, diagnosis, debugging, testing, iteration etc
It assumes the user has the right goal

- Persuasive design is a field of HCI that addresses goal formation
- Applications:
  - Reduce energy consumption
  - Promote exercise
  - Manage diet and nutrition
  - Smoking cessation
- May include “nudge” to account for biases
  - But paternalistic / patronising
It assumes the user knows what the goal is

- Not true when the purpose is an experience (third wave HCI)
- Not true in “exploratory design”

- More attention to this later in the course
- Some problems can’t be decomposed into actions
- Sometimes actions have side effects
Wicked problems

Including material provided by
Steven Tanimoto
A Wicked Problem:

Slowing climate change
More Wicked Problems

• Stopping the spread of antibiotic-resistant diseases
• Halting nuclear proliferation
• Ending homelessness in Cambridge
• Avoiding species extinction
• Colonizing Mars
1. There is no definitive formulation of a wicked problem
2. Wicked problems have no stopping rule
3. Solutions to wicked problems are not true-or-false, but good-or-bad
4. There is no immediate and no ultimate test of a solution to a wicked problem
5. Every solution to a wicked problem is a “one-shot operation”; because there is no opportunity to learn by trial-and-error, every attempt counts significantly
Rittel-Webber Characteristics 6-10 of 10

6. Wicked problems do not have an enumerable (or an exhaustively describable) set of potential solutions, nor is there a well-described set of permissible operations that may be incorporated into the plan.

7. Every wicked problem is essentially unique.

8. Every wicked problem can be considered to be a symptom of another problem.

9. The existence of a discrepancy representing a wicked problem can be explained in numerous ways. The choice of explanation determines the nature of the problem's resolution.

10. The planner has no right to be wrong.
Challenge problem: Moodle

Who is the user: You or me?
   For both Uls ... and PLs!

“Student” or “Student observer”? 