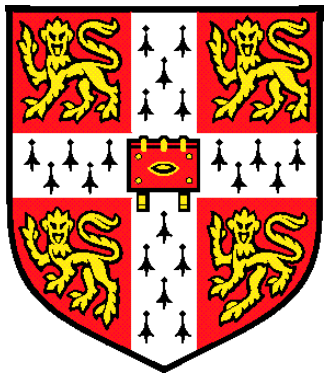
Usability Issues

in mixed initiative visual analytics

Alan Blackwell
Advait Sarkar

University of Cambridge



Outline

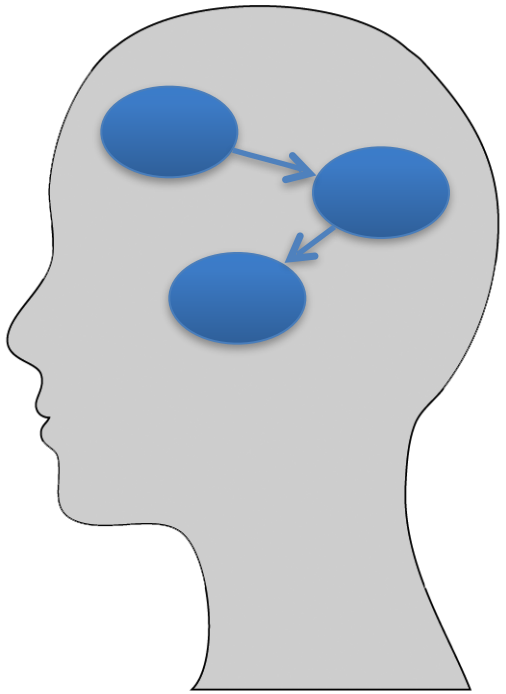
- Defining the class of system
- Mental models and user models in mixed initiative systems
- Cognitive factors in mixed initiative usability
  - Perception of control
  - Investment of attention
- Design patterns for structuring analytic interaction
- Open questions

The class of system (as intersection set)

- Intelligent Discovery Assistants (research)
  - CITRUS (Wirth et al 97)
  - AIDE (St Amant & Cohen 98)
  - IDEA (Bernstein et al 2005)
  - HAMB (Livingston et al 2001, full autonomy)
  - (...)
- Visual Analytics (commercial)
  - Omniscope
  - Tableau
  - Spotfire
  - Qlikview

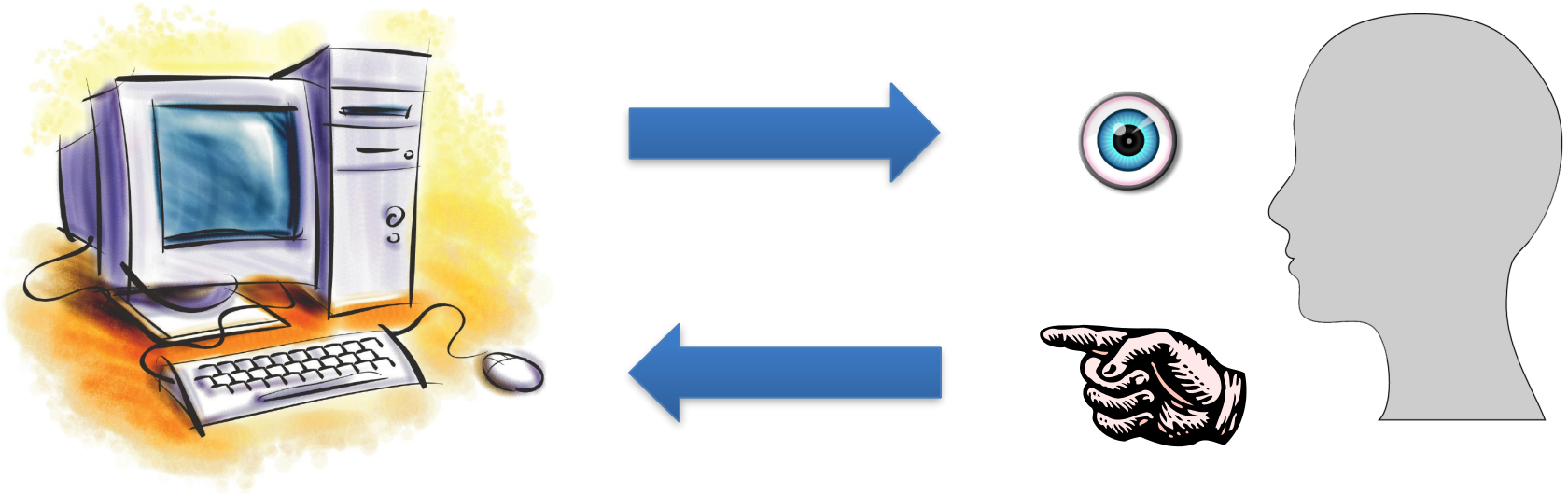
User models ≠ Mental models

Cognitive ergonomics as execution/evaluation



Mental model

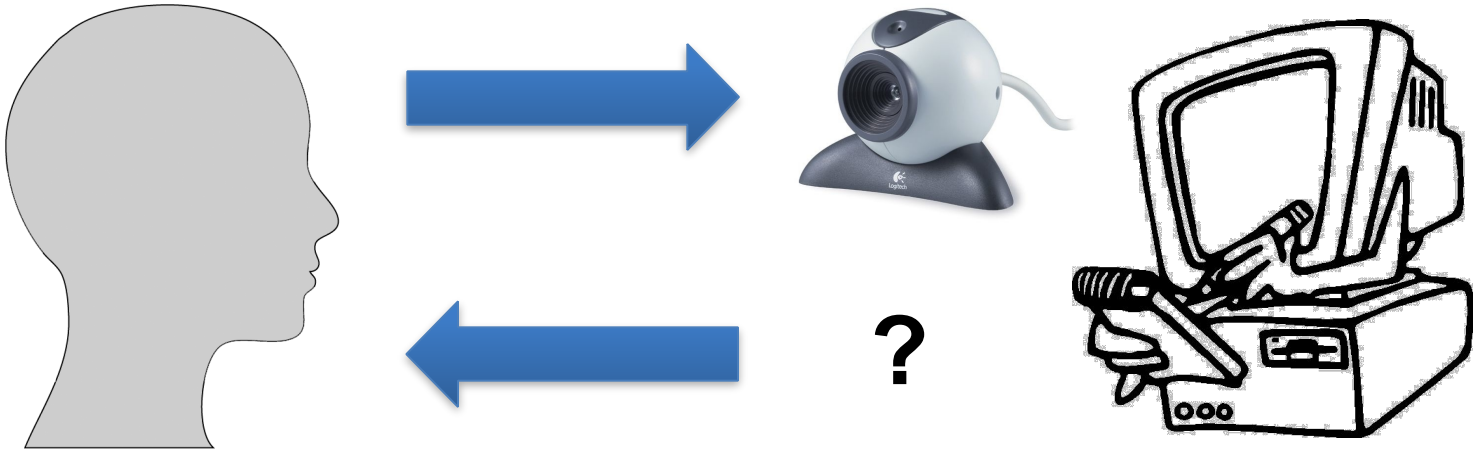
What is going on inside that thing?



How can I get it to do what I want?

User model

What is going on inside that thing?



How can I get it to do what I want?

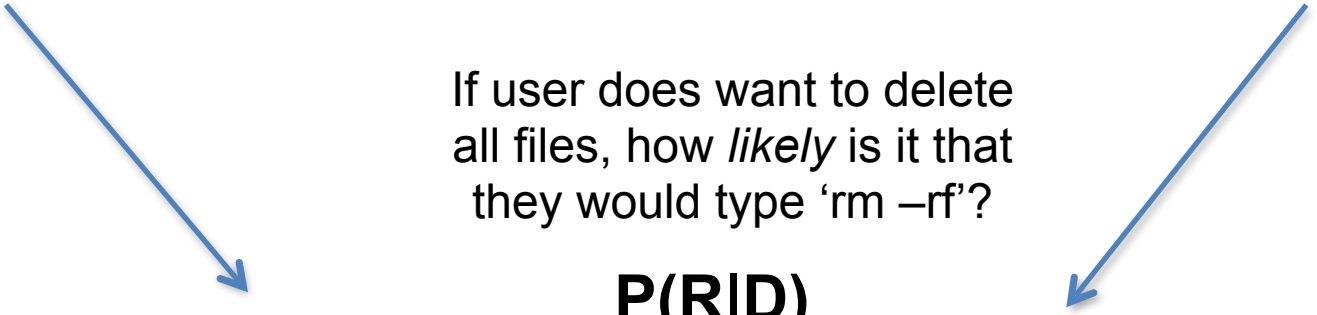
A probabilistic view of user interaction

- Machine:
  - I know how to do several things.
  - I wonder which one the user wants me to do?
- User:
  - This machine can do a whole bunch of stuff.
  - What is most likely to make it do the right stuff?
- Machine:
  - I think the user has made a mistake
- User:
  - I think the machine has made a mistake

Bayesian inference inference of user intention

Probability that user wants to delete all files, given that they just typed ‘rm –rf’

(Prior) probability that user wanted to delete all files *before* we saw this.



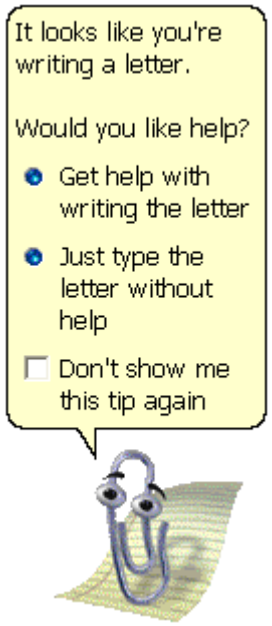
If user does want to delete all files, how *likely* is it that they would type ‘rm –rf’?

$$P(D|R) = \frac{P(R|D)}{P(R)} P(D)$$

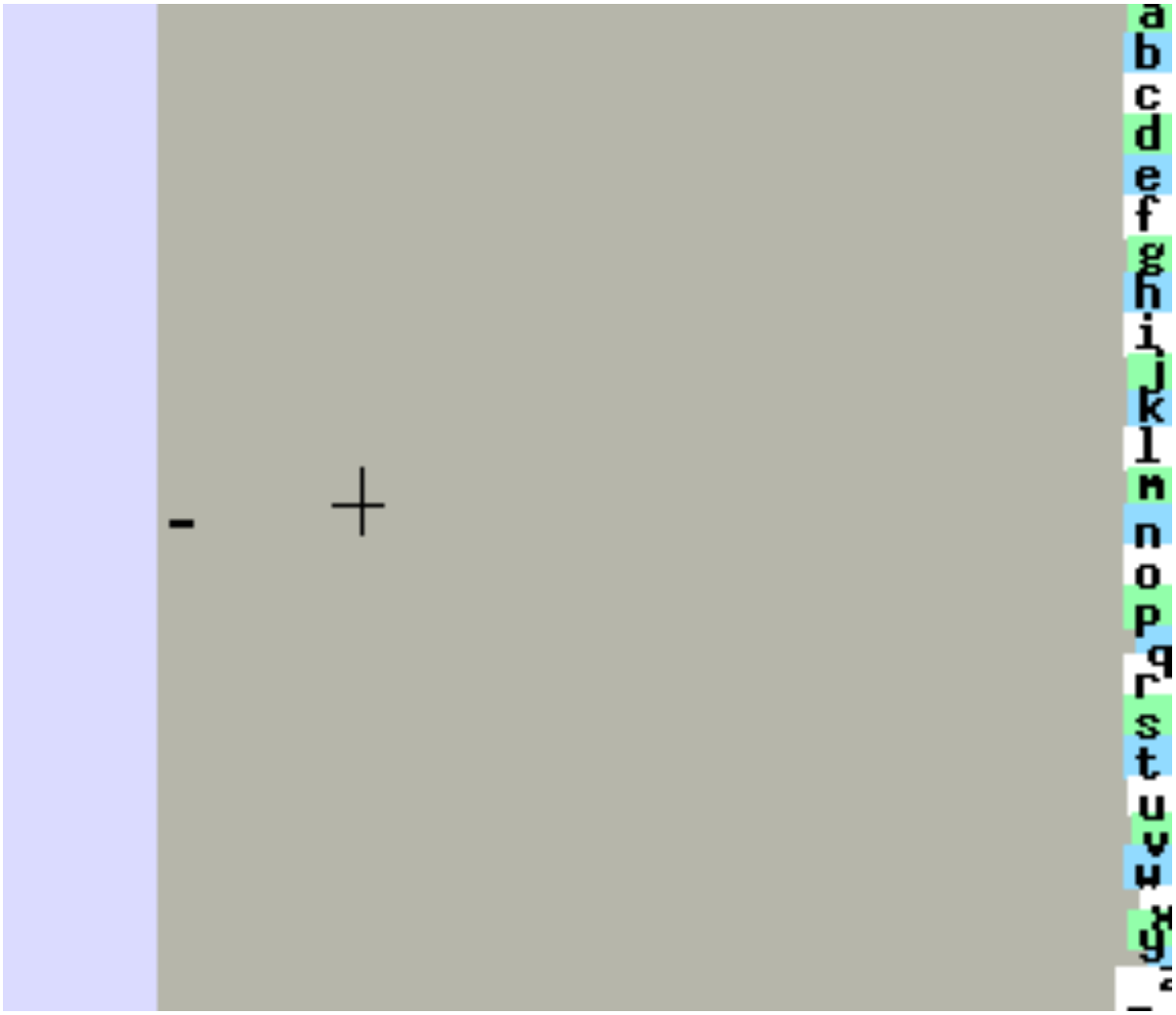
What is the probability user would type ‘rm –rf’, under all possible hypotheses?

D: User wants to **D**elete all their files
R: User has typed ‘**r**m –rf’

The case of Clippy



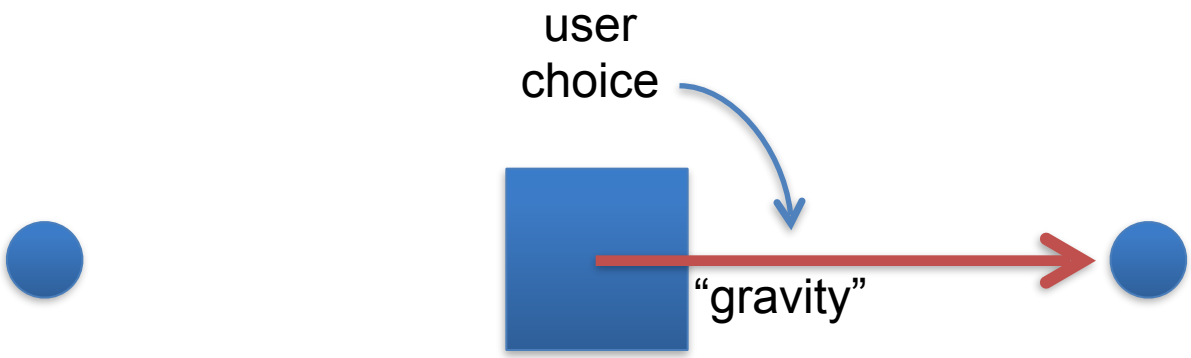
Dasher as a user model

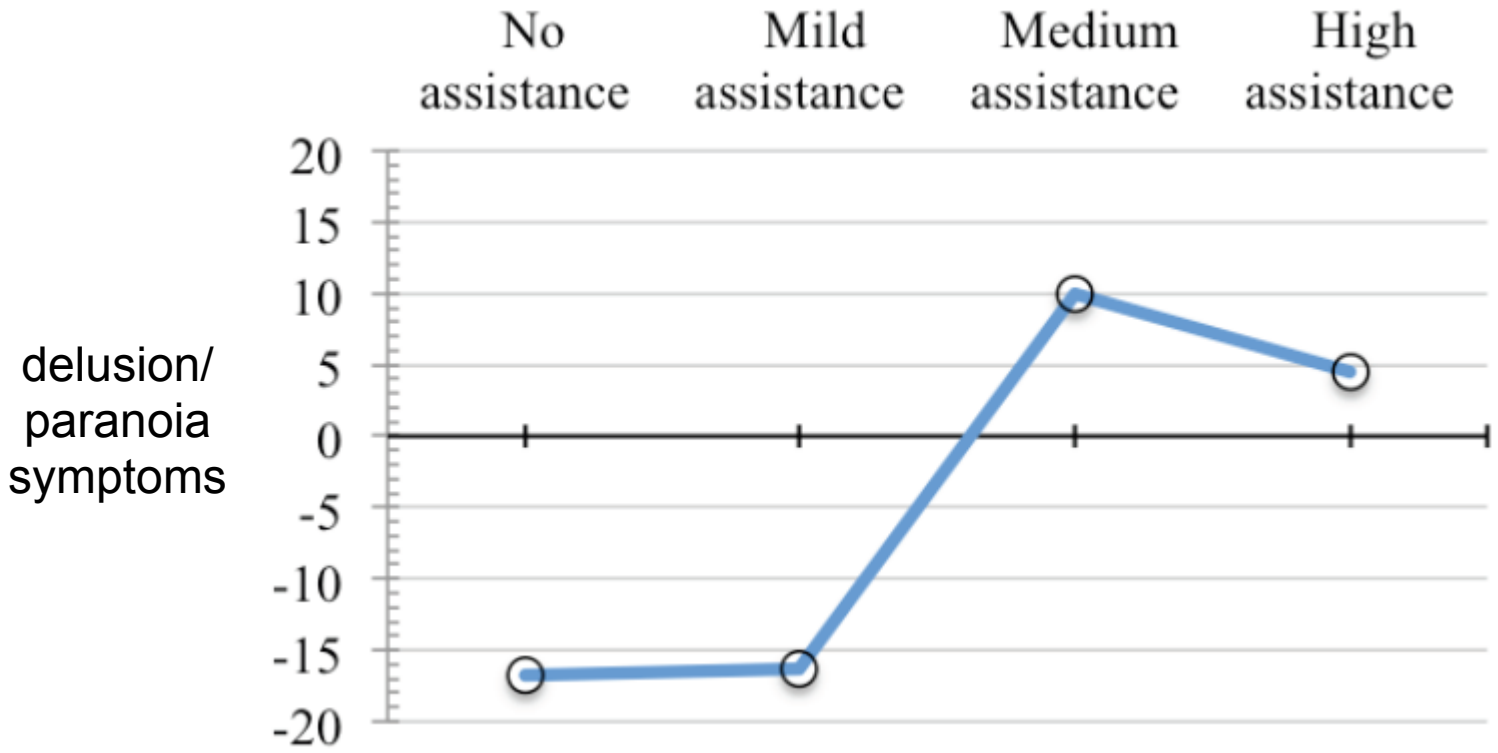


Cognitive Factors

Helping the user with “semantic pointing”

Perception of control





Attention Investment and abstraction use

- In using computers to get work done, attention (not information) is the scarce resource.
  - Mixed initiative systems reduce attention by automating operations
- But configuring the automation takes attention
  - In data analytics, this is an investment, and the payoff is reduced cost relative to manual analysis
  - Not all costs are investments, e.g. reading flashing advertisements that appear while I'm doing something else.
- There is a risk that the cost will be lost if the program/model doesn't work.

Investment parameters

- The economics is based on attention units and probabilities:
  - Cost = attention units to get the analysis done (presumably doing the analysis has value, but this is external to the theory).
  - Investment = cost if there's a potential pay-off in reduced future cost.
  - Risk = probability that a future cost will be imposed as a result of the way I've chosen to spend my attention.

- Example:
  I am thinking of creating a statistical model("program"):
  - This will cost me some attention (attention units ~ time).
  - This is an investment, because I could use the model again instead of calculating things manually.
- There is a risk (probability) I'll get it wrong, and this will cost me a lot of future attention to unravel.

Motivation for use

- If we build it, they will come?
- But, will users really ever create new abstractions? Why should they?
  - It costs attention to go explore what these things are, and I need all my attention to get my work done!
  - Further, there is risk that exploring models will be a waste of attention.
- Design strategy:
  - Encourage guidance (requires attention investment, but smaller due to timely assistance)
    - At an appropriate time consistent with user's activity.
  - Reward (pay off +) the investment.

Summary

- Attention investment is a way of understanding user problem-solving behaviors on computers.
  - Fills an important need for understanding “deep” problem-solving, like programming.
  - As opposed to simple usability of controls
- Provides a coherent design-time mechanism of making informed design choices.
  - And is much cheaper than finding big problems after building a system.

Design Patterns

- Christopher Alexander's "Pattern Language"
  == Systematic Description of User Experience
  - e.g. *Light on Both Sides of a Room*
  - (not *How to Rebate a Window Hinge*)
- Interpretation activities: reading information structures
  - Search, Comparison, Sense-Making
    (**viewing** and navigating output visualisations)
- Construction activities: building information structure
  - Incrementation, Transcription, Modification, Exploratory Design
    (**authoring** with mixed-initiative tools)
- Social activities: sharing information structure
  - Illustrate a story, Organise a discussion, Persuade an audience
    (**collaborating** via mixed-initiative analytics)

Experiences of Visibility

- The information you need is visible
- The overall story is clear
- Important parts draw your attention
- The visual layout is concise
- You can see detail in context

Experiences of Structure

- **You can see relationships between parts**
- **You can change your mind easily**
- There are routes from a thing you know to something you don't
- You can compare or contrast different parts

Experiences of Meaning

- It looks like what it describes
- **The purpose of each part is clear**
- Similar things look similar
- You can tell the difference between things
- You can add comments
- The visual connotations are appropriate

Experiences of Interaction

- Interaction opportunities are evident
- Actions are fluid, not awkward
- **Things stay where you put them**
- Accidental mistakes are unlikely
- **Easier actions steer what you do**
- It is easy to refer to specific parts

Experiences of Thinking

- You don’t need to think too hard
- You can read-off new information
- **It makes you stop and think**
- **Elements mean only one thing**
- **You are drawn in to play around**

Experiences of Process

- The order of tasks is natural
- **The steps you take match your goals**
- You can try out a partial product
- You can be non-committal
- **Repetition can be automated**
- The content can be preserved

- You can extend the language
- **You can redefine how it is interpreted**
- **You can see different things when you look again**
- Anything not forbidden is allowed

Open questions / further (actual) work!

- Controlling the mix of initiatives
  - (regulating the autonomy of the interface agent)
- Addressing barriers to entry:
  - Domain knowledge
  - Representational expertise
- Visualising/navigating the analysis space
- Reconciling exploration and hypothesis testing

Questions?

Alan.Blackwell@cl.cam.ac.uk
Advait.Sarkar@cl.cam.ac.uk