Lecture 2:
The User II: People as 'Planful' (Inter-)Actors

Jon Oberlander

School of informatics

Introduction

- Is there a generic account of how users behave which is relevant to the problems of interface design?
- Can we explain why users make errors and can we design to prevent them?
- How does knowledge and expertise influence users' behaviour?
  - Plans and 'situated actions'

Norman's Planning Model

- An account of behaviour as a repeated sequence of phases
- Elaborates the recognise-act cycle of the MHP:
  - Goal formation
  - Action execution
  - Action evaluation
- Specific types of error may be associated with each phase
- Draws attention to problems in maintaining control of an action sequence
- Monotonic – ignores interleaving of tasks, dealing with interruptions

Norman's Planning Model

1. Establish a goal, an intention to act, and a task, a practical course of action that will result in the goal being achieved
2. Form a plan for the task, a sequence of system operations to be performed on system entities
3. Translate the plan into an action specification consistent with the interface 'input' language
4. Output the action specification as a sequence of lexical tokens using the system input devices

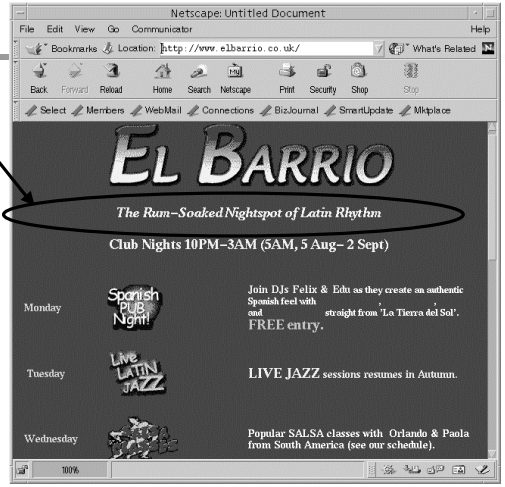
Norman's Planning Model

5. Observe the system response as e.g., visual, auditory tokens generated by output devices
6. Interpret the output in terms of changes in the system state, e.g., entity properties
7. Determine whether the new system state is consistent with what was intended
   - If not, perhaps an error has occurred
     - The plan may have been faulty
     - The plan execution may have been faulty

Example: Plan to swap 2 paragraphs

1. Move to start of 1st paragraph – options:
   - Scroll until visible
   - Move to line *nn*
2. Delete paragraph – options:
   - Delete *mm* lines
   - Mark start; move to end; cut region
3. Move to start of 2nd paragraph
4. Replace deleted paragraph
5. Delete 2nd paragraph etc.

Goal: Change www page



Plan:

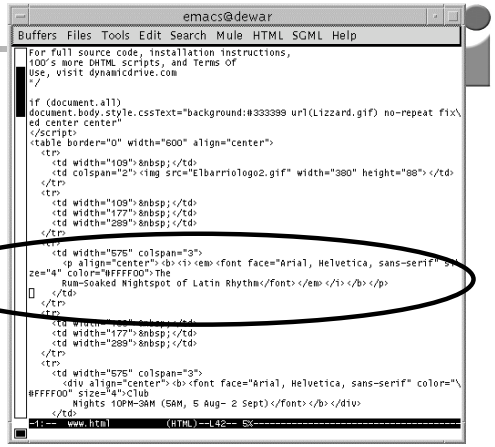
1. Find object description
2. Edit object attributes

Action specification:

- Scroll up/down
- Scroll left/right

Action execution:

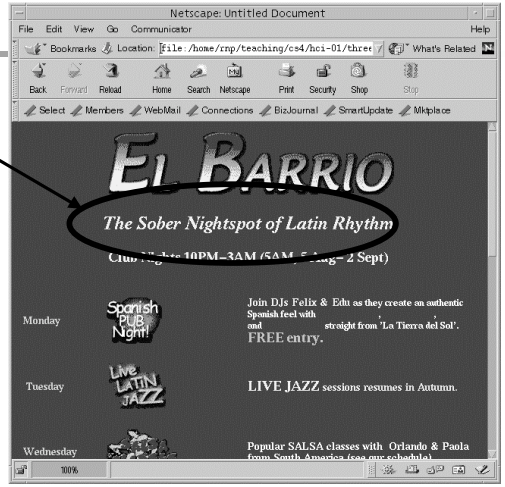
- Press Scroll keys



Observe outcome
•What has changed?

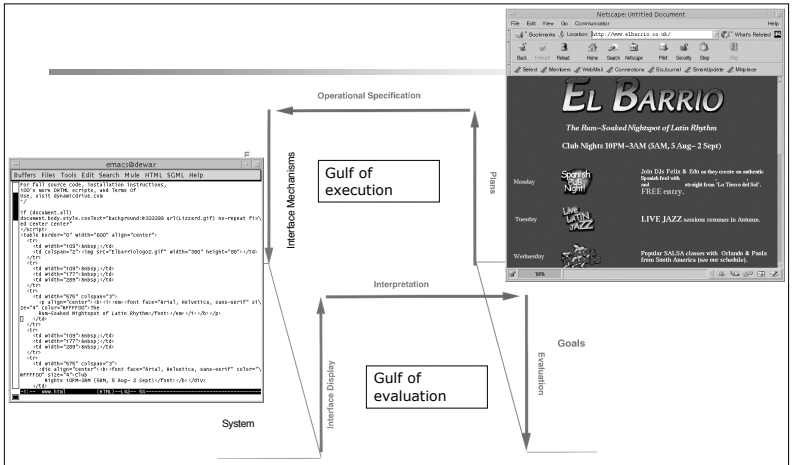
Interpret outcome
•How has it changed?

Evaluate outcome
•Is it what I intended?



Cognitive Load

- An informal measure of cognitive resources required to perform a task, e.g., working memory
- The 'gulf of execution' and 'gulf of evaluation' reflect effort of translating between the user's and the system's representation of the task:
  - System entities and operations (input language)
  - System state (output language)
- Importance of matching interface language to the user's 'task language'
  - Lengthy plans and action sequences indicate a mis-match
  - Compare high and low level programming languages



Errors

- Account for up to 30% of interaction time
- Are frustrating and sometimes dangerous
- Key usability targets
- Important to understand:
  - Kinds, of errors and their causes
  - Ways to prevent of help in their detection
  - Ways to minimise consequences

Errors of Perception

- Perception is an active system, driven by expectations as well as data
- Examples:
  - Failure to interpret information correctly --visual illusions
  - Failing to notice information – incomplete visual search

Mistakes

- Faulty 'conceptual models' may lead to incorrect plans or interpretations
- Novices lack 'how-it-works' knowledge important for detecting constraints
  - E.g., swapping paragraphs
- Systems are complex, even experts don't fully understand their behaviour
  - Distributed systems may introduce apparent non-determinism
- People's tendency to reason with too little information – to generalise inappropriately
  - E.g., language learning

Slips

- Slips occur when plans are executed incorrectly
- Two types of slip are *capture* and *position* errors
- Capture errors are failures of activation where a more frequently used command 'takes over' the execution of a less well know one
- Position errors are failures of control:
  - A step in a plan is missed out or repeated
  - Forgetting secondary goals
    - E.g., leaving your card in ATM after cash is output

Errors and Learning

- Errors can have constructive results
- Our preferred cognitive style seems to be learning by doing (or learning by watching others)
- Errors emphasise the importance of usability principles such as:
  - *Observability*
  - *Recoverability*

Plans and Situated Action

- Norman's model is an idealised version of real behaviour
- For non-expert users, planning is not extensive
- Norman's model assumes 'knowledge in the head'
  - Why plan (and learn) when we can improvise by making use of 'knowledge in the world'
- Suchman argues that, for most users, interaction, like a conversation, is not plan following, but 'situated action'

Design Guidelines

- Know the user: match the interface to the user's knowledge and tasks
- Provide prompt and relevant feedback for each action
- Provide for error diagnosis and recovery:
  - Use meaningful error messages
  - Allow for cancellation of operations in progress
  - Provide an *undo* function if feasible
  - Require confirmation of actions that have drastic (irrecoverable) consequences

Further Reading and Suggested Exercise

- Dix et al., 2nd ed, chapter 3, p. 104-9, 3rd ed, chapter 3, p. 124-130.
- Suchman, chapters 1 and 2.

- Write down a complete plan for using some familiar system, e.g., an ATM
  - How easy is it to deal with all contingencies that might arise?