COMP23420:
Introduction to User Interaction Engineering
Lecture 10

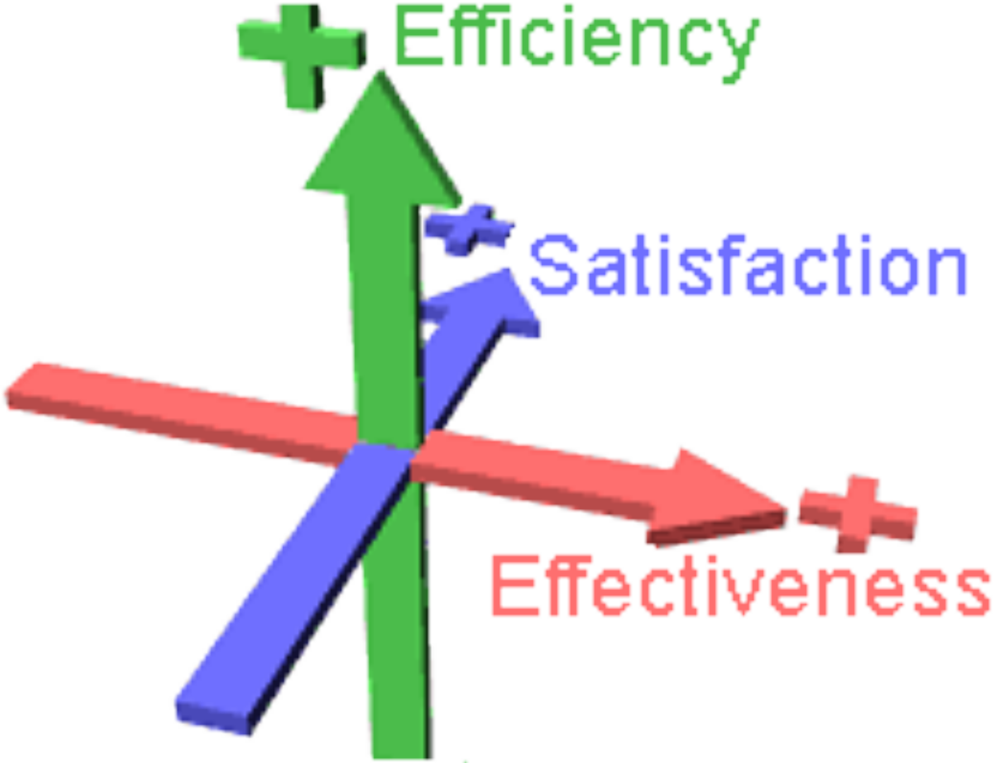
Outline

- Last week we did operational knowledge about choosing a user interface style
- This week, some of the science behind engineering human computer interaction
- A framework for talking about user interfaces
- What is it that a UI is doing?
- Justifying design choice
- Some basic user interface principles

What I Will & Will Not Tell You

- I won’t tell you how to design a good user interface
- I will tell you how to think about user interfaces
- I will tell you how to justify a design in terms of HCI
- There are lots of good books on UI design

Usability Dimensions



Usability Basics

- Allowing users to achieve a goal with efficiency, effectiveness and satisfaction
- Utility is the functionality of a system
- Utility without usability, but not *vice versa*
- Worthy, but unhelpful
- Have paradigms of good usability, e.g. GUI
- Also need theory to know why something is usable
- Really want principles to guide developers – engineering not craft

Usability Principles

a. Visibility of system status System should always keep users informed

b. Match between system and the real world System should speak the user's language

c. System functions chosen by mistake need a clear 'emergency exit'

d. Consistency and standards Avoid ambiguity

e. Error prevention

f. Recognition rather than recall

g. Flexibility and efficiency of use

h. Aesthetic and minimalist design

i. recognize, diagnose and recover from errors

j. Help and documentation

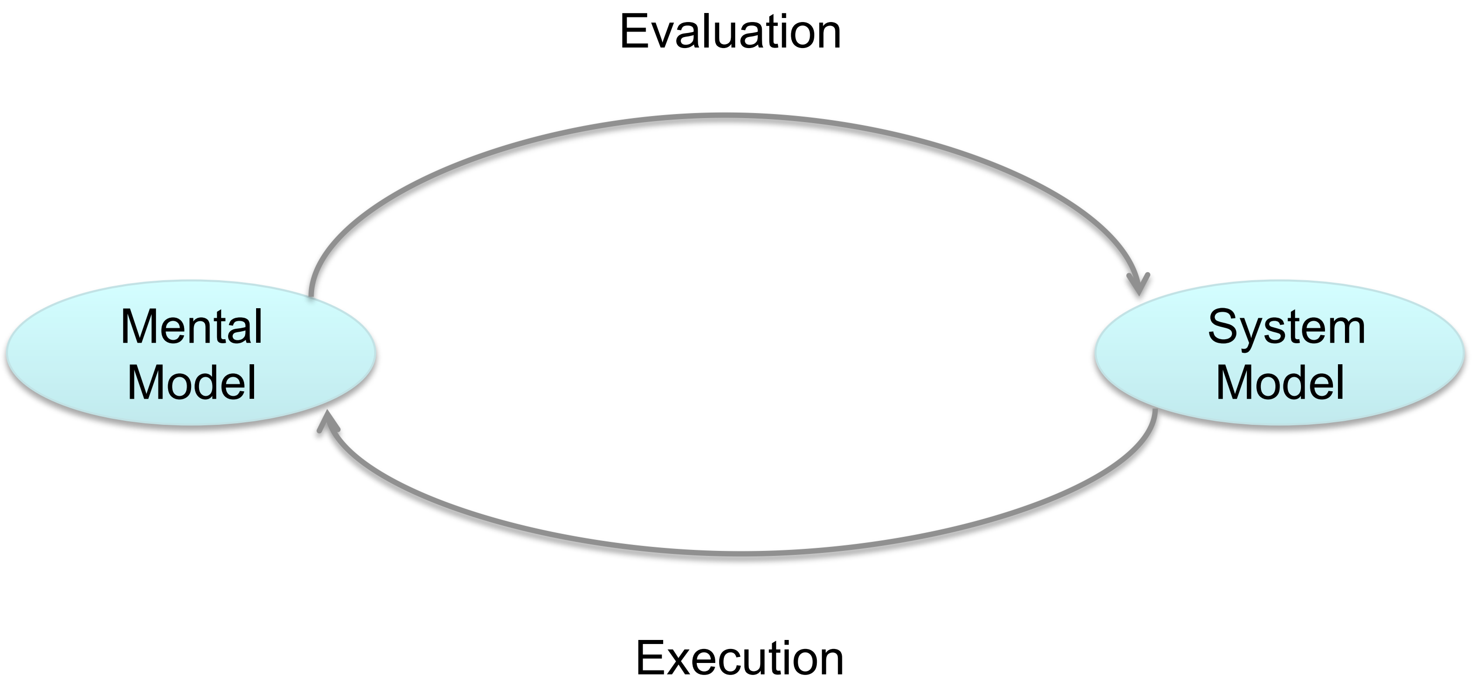
The Human

- Has mental models of goals and intentions
  The human recieves information; forms intentions and executes actions with tools in the world
  The human “information processor” has limitations
  Brilliant at association and inference
  Limited memory: The famous 7 plus or minus two
  Physical limitations – finger span, tiredness

The Computer

- Very fast
- Good memory
- Good at long, repetitive task (at which humans are poor)
- Rubbish at inference and association
- Not very fault tolerant
- A very different view of the world to a human

1. Cycle of Execution and Evaluation

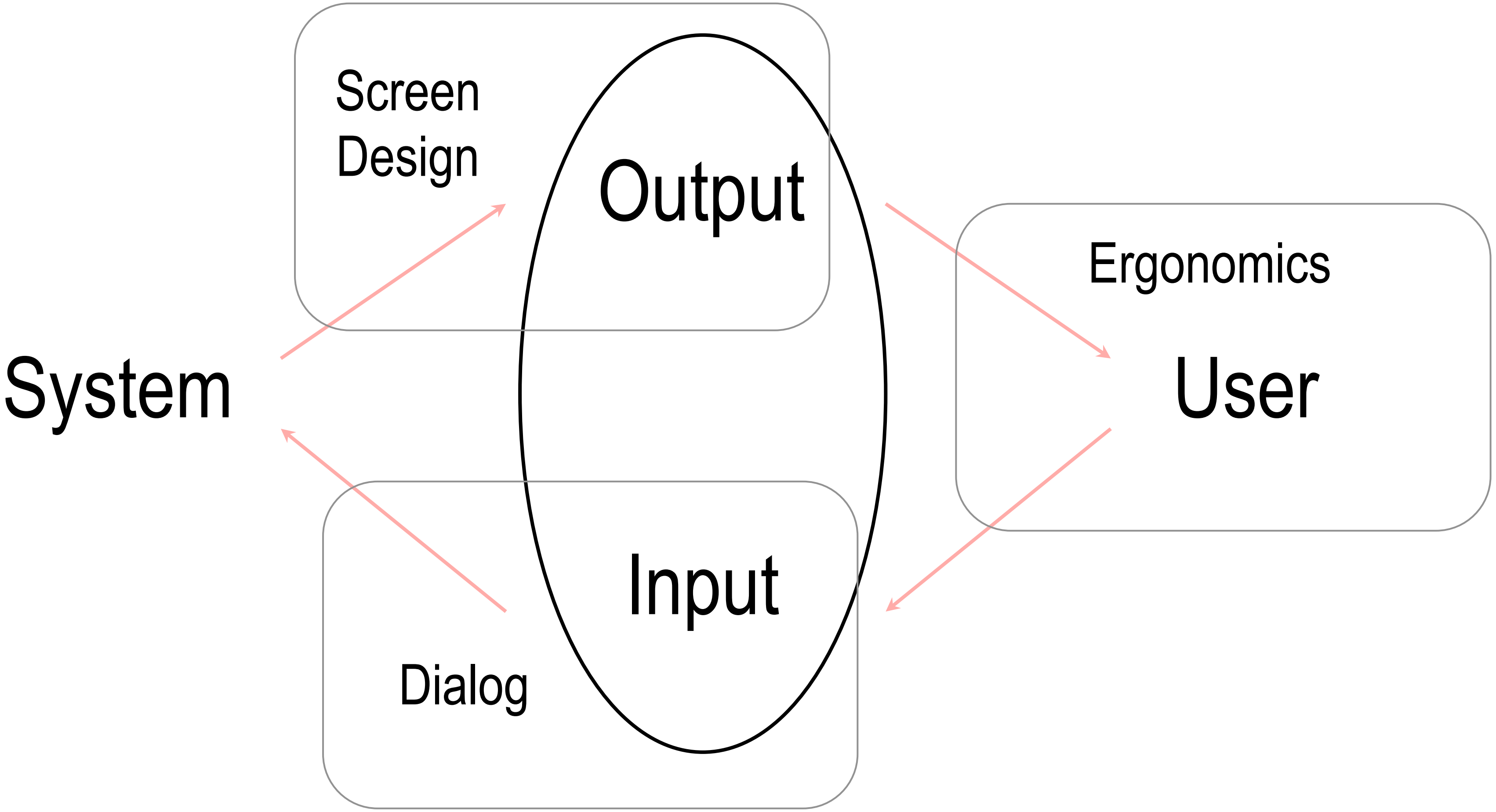


By Don Norman

The job of the user interface

- A user has his or her model of a task
- The system has a model of the task
- The UI mediates between the two
- As a software engineer your job is to help form the right model in the software
- The UI's job is to help convey that model to the user
- Ideally, they should all match

HCI & Interaction Framework



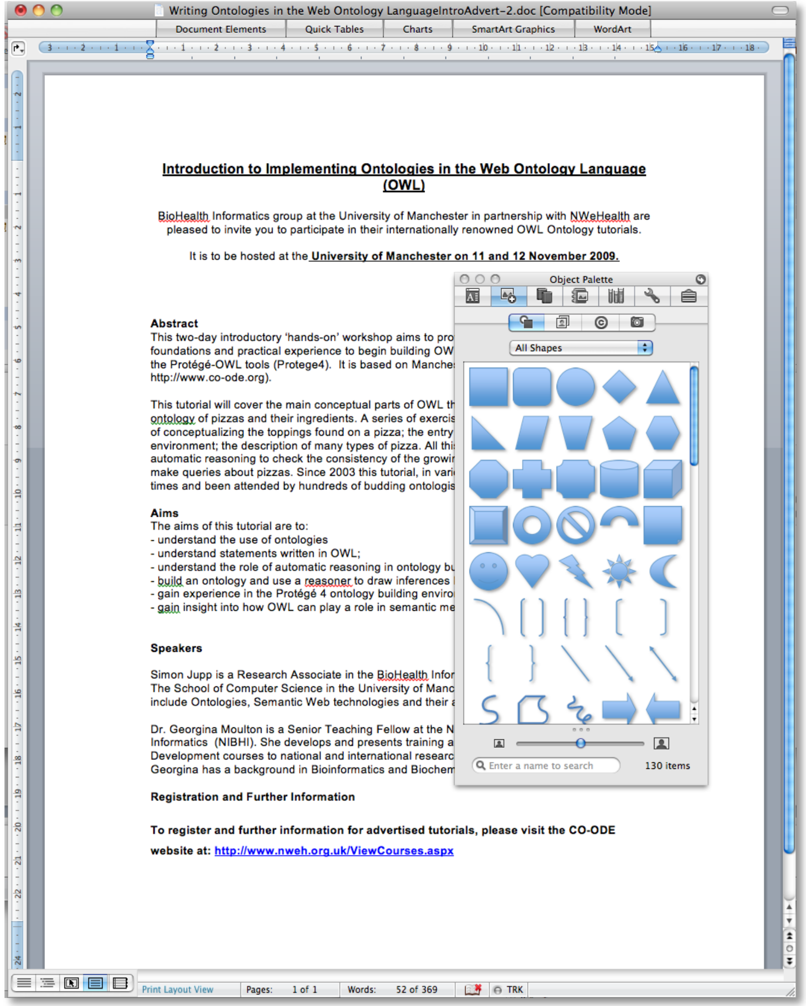
Talking About HCI

- Presentation: How the system renders state and allows the user to evaluate state and alteration to the state
- Observation: What the user notices of the presentation; Can he/ she see what they need to?
- Articulation: Expression of a user’s execution plan
- Performance: the system’s execution of a plan, the results of which are presented to the user

Frameworks and HCI

- Framework used to co-ordinate HCI issues
- Ergonomics: Physical aspects of input and output, including layout – the user side
- Dialogue Design: Task articulation and performance – the system side
- Rendering State: Presenting system state for evaluation – user and system sides

Ergonomics ofTypical GUI



Introduction to Implementing Ontologies in the Web Ontology Language (OWL)

BioHealth Informatics group at the University of Manchester in partnership with NWeHealth are pleased to invite you to participate in their internationally renowned OWL Ontology tutorials.

It is to be hosted at the **University of Manchester on 11 and 12 November 2009.**

Abstract
This two-day introductory 'hands-on' workshop aims to pro
foundations and practical experience to begin building OW
the Protégé-OWL tools (Protege4). It is based on Manche
http://www.co-ode.org).

This tutorial will cover the main conceptual parts of OWL th
ontology of pizzas and their ingredients. A series of exercis
of conceptualizing the toppings found on a pizza; the entry
environment; the description of many types of pizza. All thi
automatic reasoning to check the consistency of the growi
make queries about pizzas. Since 2003 this tutorial, in vari
times and been attended by hundreds of budding ontologis

Aims
The aims of this tutorial are to:
- understand the use of ontologies
- understand statements written in OWL;
- understand the role of automatic reasoning in ontology bu
- build an ontology and use a reasoner to draw inferences
- gain experience in the Protégé 4 ontology building enviro
- gain insight into how OWL can play a role in semantic me

Speakers

Simon Jupp is a Research Associate in the BioHealth Infor
The School of Computer Science in the University of Manc
include Ontologies, Semantic Web technologies and their a

Dr. Georgina Moulton is a Senior Teaching Fellow at the N
Informatics (NIBHI). She develops and presents training a
Development courses to national and international researc
Georgina has a background in Bioinformatics and Biochem

Registration and Further Information

To register and further information for advertised tutorials, please visit the CO-ODE website at: http://www.nweh.org.uk/ViewCourses.aspx

Recall vs Recognition

- With the $ prompt “what do I do now?”
- The user has to recall actions to perform and the syntax to use
- With the WIMPS GUI one only has to “recognise” what to do
- I go to “File” for doing file action
- Inside the file menu I see “Print” and then all its options
- Otherwise I have to remember lots of Unix options

Modelessness

- A mode places interpretations upon user’s actions: What a keystroke means in a pparticular situation
- Modelessness is an ideal property of a user interface i.e. the user should be able to do what they want whenever they want
- Modelessness means the computer interface should not have distinct modes that restrict the user's actions depending on the mode (s)he is in.
- Not separate modes for drawing, writing, tables, etc.
- Ideally the use of modes is to be avoided, because a mode typically restricts the operations that the user can perform

Mini Modes

- However one problem presented by Modelessness is that the user cannot cope with everything at once.
- Some use of modes is therefore acceptable:
  - Long-term modes modes, e.g. doing word processing as opposed to graphics editing
  - Short term "spring-loaded" modes - which continue while the user does something continuously, e.g. while the mouse button is held down
  - Alert modes modes - require user to rectify an unusual situation before proceeding
  - Modes that emulate a real-life situation that is itself modal - e.g. use of different graphics tools

More Mini Modes

- Modes that change the attributes of something rather than its behaviour - e.g. boldface and underline modes of text entry
- Modes that block most other normal operation of the computer to emphasize the modality, as in error conditions
- There should be a clear indicator of the current mode, such as a pointer whose appearance changes according to the mode
- It should be easy to change modes, e.g. graphics pointer and palette

Learnability

- The ease with which new users can start interaction
- Ease with which users can complete a task
- Closely linked to memorability
- Ease to learn; easy to remember
- Should be able to return after long break and know how to use the tool
- Should be able to return after a short break (cup of tea) and apprehend state
- Consistency of application design helps
- Closeness to user’s model of the task helps mapping
- Also linked to *recognition*

Learning by Exploring

- Most users don’t read manuals
- Users prefer to learn by playing or exploring
- Therefore good presentation of model to user is vital
- “Walk up and use”
- Transferable skills

Ease of Translation

- Concepts of application domain need to be clear in the user interface
- Help form good *mental model*
- User needs to map easily from task language to input language
- Ease of articulation – Gulf of execution
- “Speaking the user’s language” and “natural interaction style” *cf* heuristic evaluation
- VR eases articulation by making the everyday part of input language

Predictability

- Having executed an action, does the user know what will happen next?
- Pressing “OK” in a dialogue; selecting a file in the “open” dialogue
- “What will happen next?”

The Gas Hob

Consistency

- Consistency with task: External consistency
- Aids learning through transfer
- Ergonomic issues; grouping, frequency, etc.
- Arbitrary consistency: Internal consistency
- When there is no meaning to the mapping or consistency with task
- Light switches, hot and cold taps,…
- Do the same things in the same way

Reachability

- Can the user *reach* all parts of the system?
- Does the UI map from all parts of task model to system model?
- Can the system do things that I cannot ask it to do?
- Turning off the paperclip
- Stopping the department’s Unix system asking if I wish to delete a file?

Translating Input

- Input language translated to the system’s core language
- Can the input reach all the states of the system necessary?
- Video remote controls and power buttons
- Remote control’s input language cannot reach *off* state of system
- Match task analysis or activity diagrams to use cases and class/ collaboration diagrams
- Small cost to user, larger cost in implementation

Ease of Evaluation

- Performance of task transforms state
- Translate state from core to output language
- Must preserve state of system attributes in terms of domain concepts as presented by output language
- Output language often limited in expressivity
- Video simply limited in size – difficult to see context in documents etc.
- Results of file copy in command system

Heuristic Evaluation

- A set of heuristics (rules of thumb) developed by Jakob Nielsen and Rolf Molich
- http://www.useit.com/papers/heuristic/heuristic_evaluation.html
- Each heuristic used to critique an interface
- A set of **independent** experts use the heuristics
- Problems found following a Poisson distribution – 5 experts find about 75% of problems
- Usability questions used to guide and stimulate
- Essentially a check list

Nine Principles

1. **Simple and natural dialogue**: simple means no irrelevant or rarely used information, natural means an order that matches the task.
2. **Speak the user's language**: use concepts from the user's world; don't use system-specific engineering terms.
3. **Minimize user memory load**: don't make the user remember things from one action to the next: leave information on the screen until it is no longer needed.
4. **Be consistent**: action sequences learned in one part of the system should apply in other parts.
5. **Provide feedback**: let users know what effect their actions have on the system.
6. **Provide clearly marked exits**: if users get into part of the system that doesn't interest them, they should be able to get out quickly without damaging anything.
7. **Provide short cuts**: help experienced users avoid lengthy dialogs and informational messages they don't need.
8. **Good error messages**: let the user know what the problem is and how to correct it.
9. **Prevent errors**: whenever you discover an error message, ask if that error could have been prevented.

The main point

- The UI maps between human, system model and back again
- Creating the right model on each side
- Easing transation between the two
- Taking into account the characteristics of each side

The January exam

- 90 minutes
- On paper
- 40 multiple choice questions worth 20 marks
- A choice between two longer, 20 mark multi-part questions,
- All part of the semester one course are covered
- There will be a revision session just before the exam.

Reading

- First three chapters of HCI by Dix Finlay, Abowd and Beale
- Designing the User Interface: Strategies for Effective Human-Computer
- Interaction; Ben Shneiderman, Catherine Plaisant. Addison Wesley; 4 edition, 2004. ISBN-10: 0321197860; ISBN-13: 978-0321197863