MANCHESTER 1824

The University of Manchester

COMP23420: Introduction to Human Computer Interface Design Lecture 9

Outline

- Why human computer interface engineering in this course
- Styles of user interface
- Choosing a user interface style
- Workshop five
- Next week, talking about HCI

Why Human Computer Interface Engineering?

- User interfaces are part of software
- User interfaces need engineering too
- You can have the best functionality ever…
- … but if you can't deliver it, it's useless
- The design of user interfaces has it's own considerations
- We can use scientific principles (as well as economics and social organisation) to solve a human computer interaction problem; it's engineering

- And that's what we'll start doing here.

When to Build your User Interface

- *Typically* the last thing and not enough time
- *Should* do it early on…
- Can design a UI as soon as the functionality is known – or to help find the functionality
- Early prototyping; keeping it Agile
- Always having a working version of whatever you have
- Also helps force requirements questions

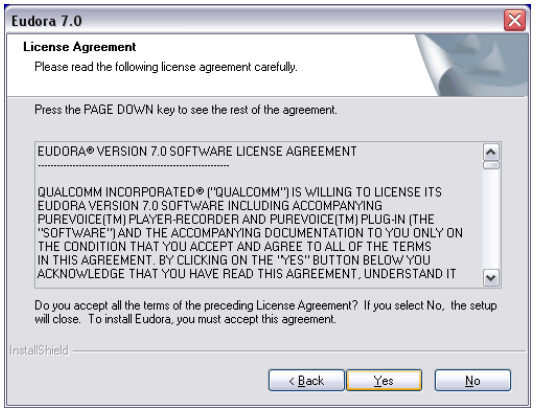
WIMPS

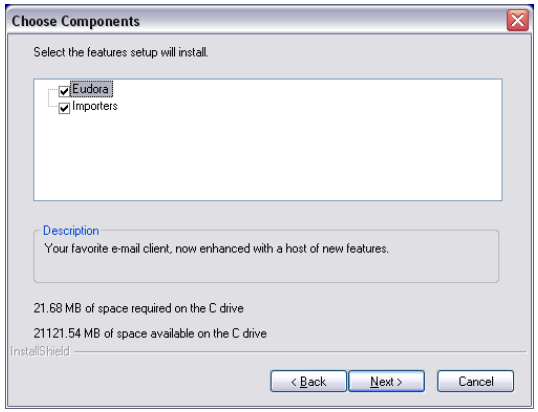
- Windows, Icons, Menus, Pulldowns and Selection
- The user interface widgets that you commonly use
- There are many styles for deploying these widgets in a UI
- Some are not based on WIMPS
- The user interface designer's skill is choosing a design that affords the functionality
- A different skill from programming
- Some people would claim disjointness in the two skill sets

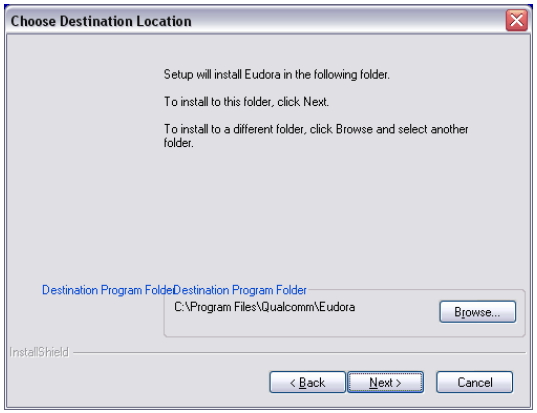
Styles of User Interface

- UI means by which a user accesses the functionality of a system + the means by which system state is presented
- Different functionalities imply different styles of user interface
- The form in which WIMPS are deployed gives the style
- It implies a style of interaction

Wizard Interface



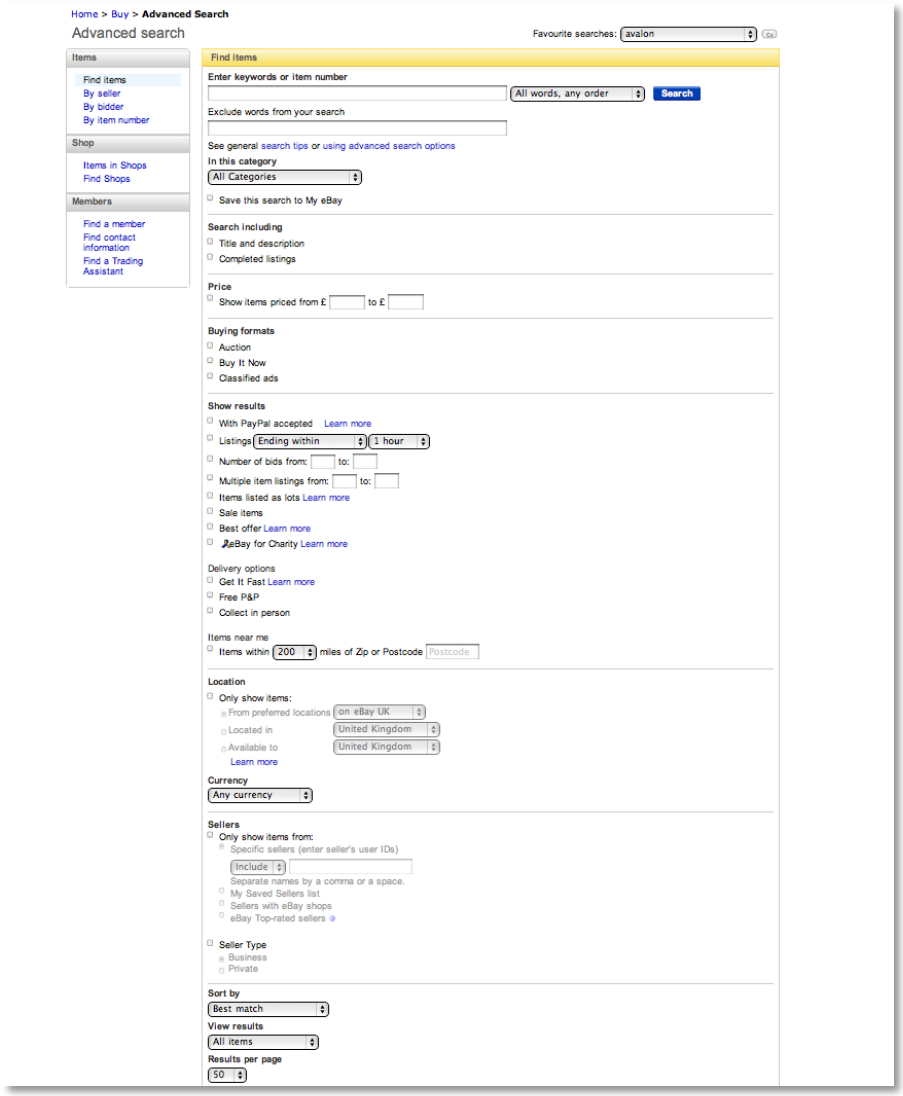




Choosing wizards

- Usually single, narrowly defined goal
- Defined start point
- Restricted set of known steps
- Usually in a defined order(s)
- Set of known options
- Ability to move back and forth and change options
- One choice affects what is offered for another
- Typical of an installation process

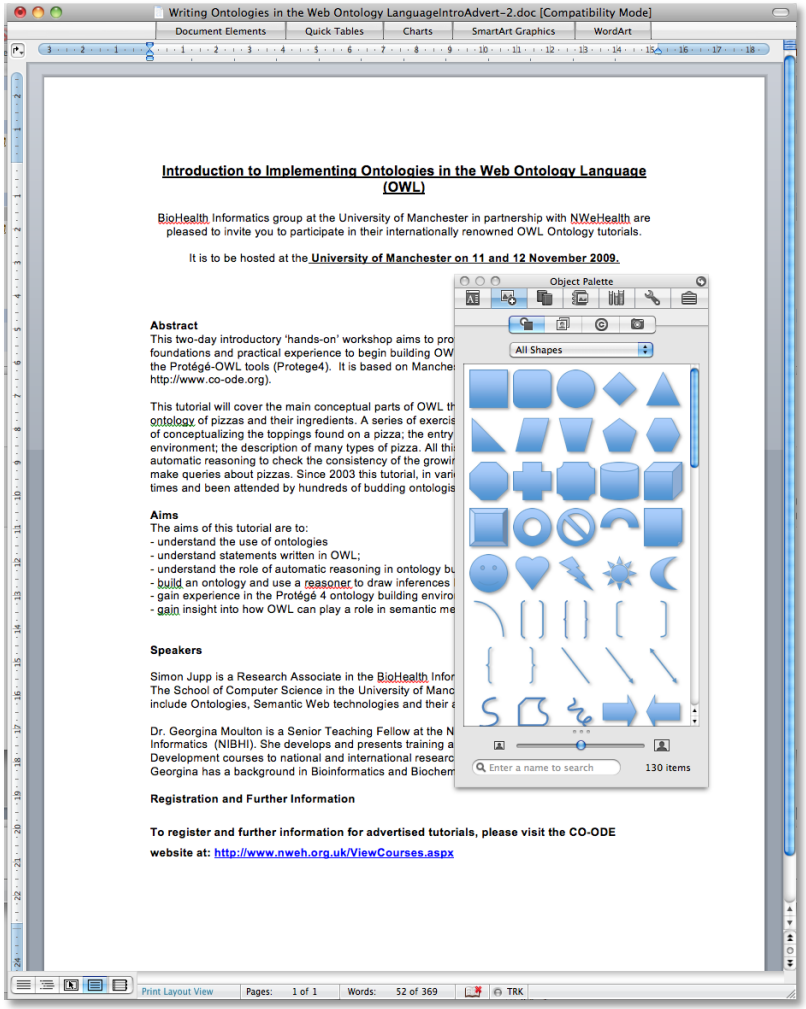
Form Fill-in


Home > Buy > Advanced Search
Advanced search
Favourite searches: avalon
Items
Find items
By seller
By bidder
By item number
Shop
Items in Shops
Find Shops
Members
Find a member
Find contact information
Find a Trading Assistant
Find items
Enter keywords or item number
All words, any order
Search
Exclude words from your search
See general search tips or using advanced search options
In this category
All Categories
Save this search to My eBay
Search including
Title and description
Completed listings
Price
Show items priced from £ to £
Buying formats
Auction
Buy It Now
Classified ads
Show results
With PayPal accepted Learn more
Listings Ending within 1 hour
Number of bids from: to:
Multiple item listings from: to:
Items listed as lots Learn more
Sale items
Best offer Learn more
eBay for Charity Learn more
Delivery options
Get It Fast Learn more
Free P&P
Collect in person
Items near me
Items within 200 miles of Zip or Postcode Postcode
Location
Only show items:
From preferred locations on eBay UK
Located in United Kingdom
Available to United Kingdom
Learn more
Currency
Any currency
Sellers
Only show items from:
Specific sellers (enter seller's user IDs)
Include
Separate names by a comma or a space.
My Saved Sellers list
Sellers with eBay shops
eBay Top-rated sellers
Seller Type
Business
Private
Sort by
Best match
View results
All items
Results per page
50

Choosing a Form-fill in Style

- Again, set of known steps and known options
- No defined start point
- I might want to start anywhere (no defined ordering)
- Choosing Specifying a package holiday: Start point; end point; dates; hotels; tours; etc.
- I might wish to start at any point
- One choice affects another

Direct Manipulation Interface



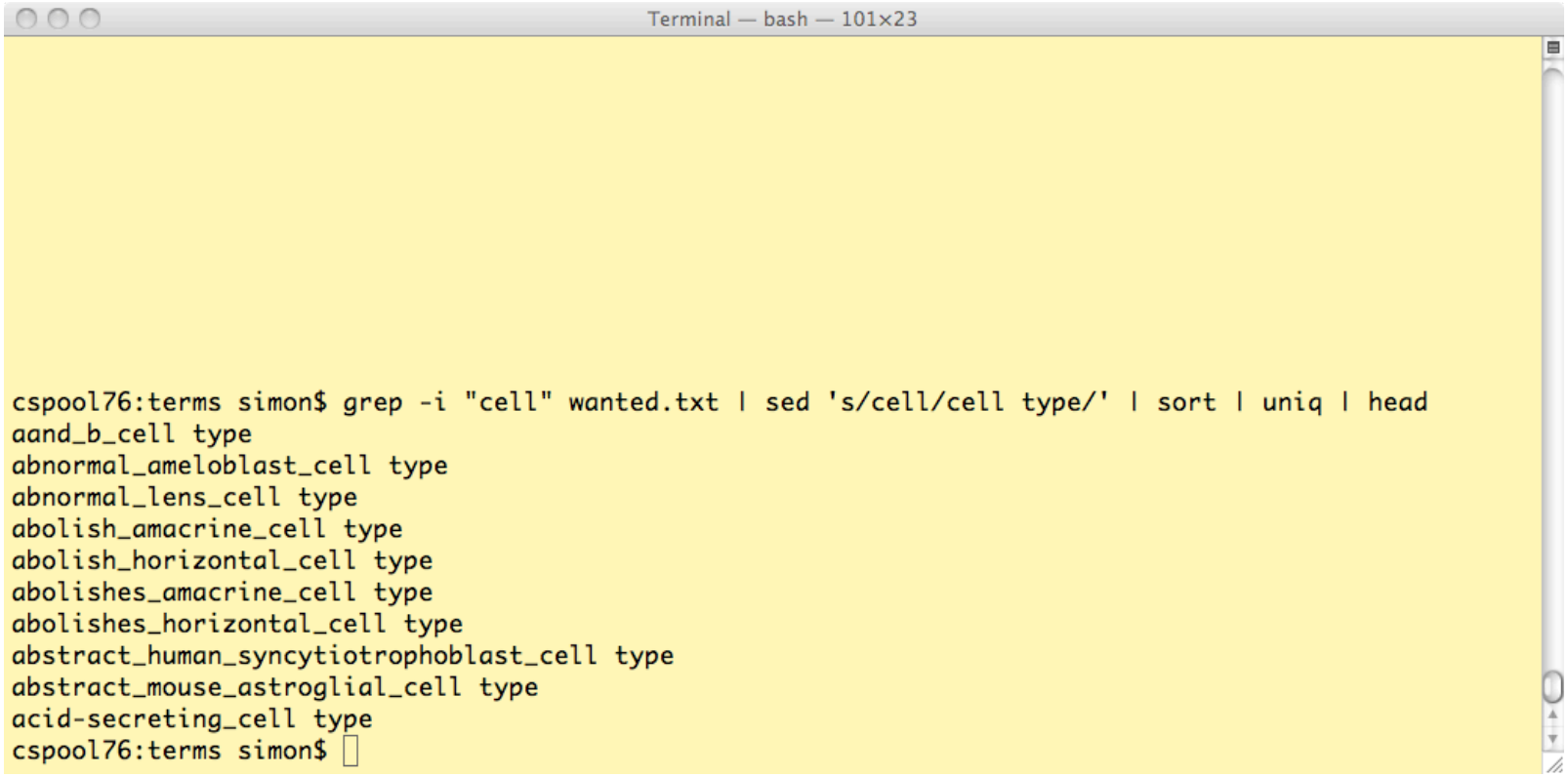
Direct Manipulation

- Humans work by manipulating things both mentally and physically
- A user reaches for what he/she wants, uses and puts it down again
- Humans also wish to see the results of their actions
- Dragging a file to the printer; putting files in the bin; making words bold; drawing lines etc.
- “hey you do that”
- Lots of functionality available all the time

Choosing Direct Manipulation

- Wide array of functionality
- No real ordering in use of functionality
- Where recognition rather than recall can dominate
- Where WYSIWYG feedback is appropriate
- Remember, DM UI often contain many kinds of UI styles

Command line interface



Choosing a Command-Line Style

- Typically not for everyday users
- Very good for mass action
- “mv *.txt ../other-folder”
- Very good for constructing arbitrary, complex actions
- Often very fast in certain tasks for the power user

Batch Processing

- Cron

```
#minute hour    mday    month   wday    who     command
0       10      *       *       *       root    /usr/sbin/ntpdate -b pool.ntp.or
0       22      *       *       *       root    /usr/sbin/ntpdate -b pool.ntp.or
55      2       *       *       1,4     root    sync; /usr/syno/bin/synousbdisk
5       1       *       *       1,4     root    /usr/syno/bin/synonetbkp -a "DS1
13      *       *       *       *       root    /var/packages/Webalizer/target/b
5       1       *       *       1,4     root    /usr/syno/bin/synonetbkp -a "DS2
5       1       *       *       1,4     root    /usr/syno/bin/synonetbkp -a "DS2
5       1       *       *       1,4     root    /usr/syno/bin/synonetbkp -a "DS2
5       1       *       *       1,4     root    /usr/syno/bin/synonetbkp -a "DS2
5       1       *       *       1,4     root    /usr/syno/bin/synonetbkp -a "DS2
5       1       *       *       1,4     root    /usr/syno/bin/synonetbkp -a "DS2
5       1       *       *       1,4     root    /usr/syno/bin/synolocalbkp -a "T
5       1       *       *       1,4     root    /usr/syno/bin/synolocalbkp -a "T
5       1       *       *       1,4     root    /usr/syno/bin/synolocalbkp -a "T
5       1       *       *       1,4     root    /usr/syno/bin/synolocalbkp -a "T
5       1       *       *       1,4     root    /usr/syno/bin/synolocalbkp -a "T
5       1       *       *       1,4     root    /usr/syno/bin/synolocalbkp -a "T
5       1       *       *       1,4     root    /usr/syno/bin/synolocalbkp -a "T
5       1       *       *       1,4     root    /usr/syno/bin/synolocalbkp -a "T
~
~
~
~
- /etc/crontab 1/20 5%
```

Choosing a batch Processing Style

- A set of repeated steps
- Need to be run repeatedly in the same way
- “A batch” of commands
- Don’t necessarily need to interact as the process runs
- Bbut do need feedback like logs
- Still an interaction style

Workshop Five

- Design a UI for the HTV system case study
- Use the HTV system use cases
- Sketch the UI and justify choice
- Bring the HTV scenario; use cases and glossary.

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
- Only as good as the person doing the evaluation

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.

You have been watching…

- Some absolute basics of user interface design
- Different design styles to be chosen
- Choice based on an understanding of interaction patterns
- Comes from understanding the business process