HCI Online Material

Chapter 1

Online Resources

A train ticket-booking machine simulator used in the south of England.
http://www.uclic.ucl.ac.uk/projects/tvm/simulation/index.html

Prof Thimbleby's research paper about the ticket machine simulator.
http://www.uclic.ucl.ac.uk/projects/tvm/

Activity 1.1 - Train Ticket Machine Simulator

Follow this link: http://www.uclic.ucl.ac.uk/projects/tvm/simulation/index.html

This is a simulation of a train ticket-booking machine as used in the south of England. Press the 'Station context' and 'Front panel' buttons to see photos of the machine on a station platform and a close- up of the front of the machine.

The simulation is based on the main interactive part of the machine where the user selects their destination and ticket type (the part of the machine with a big '1' pointing to it on the front panel photo).

Click the check box marked 'Prefer photos' to remove the tick (so that you don't get photos) and it is probably best to enlarge the window as much as possible.

Press 'Cancel' to start the simulation working. The arrow buttons down the left and right hand sides act like the buttons on the actual machine, and the central frame shows the display. The idea is to press the arrow that points to the operation you want to perform. So if you wanted a ticket to Cambridge you should press the fourth button from the top on the left.

Take a few moments to explore the simulation to get used to it.

Feedback on Activity 1.1 - Train Ticket Machine Simulator

This simulation was researched and programmed by Prof. Harold Thimbleby of the University College Interaction Centre to demonstrate how difficult simple, everyday gadgets are to use.. Prof. Thimbleby's research paper about the machine can be accessed from the following link: but be warned; this is a high level research paper which deals with many other wider aspects of designing interactive systems that we have not dealt with yet on this course. Do not worry if you don't understand it at the moment.

http://www.uclic.ucl.ac.uk/projects/tvm/

Activity 1.2 - Booking a Train Ticket

Again follow this link:
http://www.uclic.ucl.ac.uk/projects/tvm/simulation/index.html
Use the train ticket simulator to book a ticket from Welwyn North (where the machine is) to New Southgate with a 'Network Gold card' (a Network Gold card is a discount card that you buy once a month or so which allows you discounts on all the train tickets you buy).

1. Time yourself buying the ticket.

2. Make sure that you actually get the discount for the Gold card; i.e. your ticket should cost £3.70, not £5.60.

3. If you do manage to successfully book the ticket, go back to the beginning, click the 'Ticket paper' check box (so that it is not selected) and then try again. What happens?

Feedback on Activity 1.2 -Booking a Train Ticket

How did you do?

You probably found out how to select New Southgate, but did you work out how to get a discount with your network railcard? If you did, well done!

Usually we find that it takes computing science students two or three minutes to work out how to book the ticket, usually with considerable help and advice from their friends. In a lot of cases people give up altogether and decide that it is impossible.

How you should actually do it is to How you should actually do it is select 'Other destinations' from the first screen, then press 'M-N-O' (for New Southgate) and then press 'Next Screen' a few times until 'New Southgate' is displayed. Now select '£5.30 Standard Day single', then select 'Railcard discounts', then '£3.70 Network Gold Card'. Then you can buy your ticket.

The problem that confuses many people is that on the screen titled 'Adult Standard tickets to New Southgate' you have to select '£5.30 Standard Day single' which is actually *not* the ticket you want!

If you tried with the 'Ticket paper' box unchecked this means that the machine has run out of paper to print its tickets on. But it allows you to go through the whole process of selecting your ticket, before it tells you that it cannot actually print your ticket. Why does it not tell you at the very beginning so as not to waste your time?

Also click through the buttons down the right hand side 'No fares', 'No service' etc. to see some of the other problems this machine suffers from.

Chapter 2

Activity 2.3 - Designing a New Ticket Machine

Follow this link: http://www.uclic.ucl.ac.uk/projects/tvm/simulation/index.html

This is a simulation of a train ticket-booking machine as used in the south of England. Press the 'Station context' and 'Front panel' buttons to see photos of the machine on a station platform and a close- up of the front of the machine.

You used this simulation for Activities 1.1 and 1.2 in Chapter 1. Review your notes from those Activities.

Now take some fresh sheets of paper and design a new ticket machine, which does not suffer from the problems you saw on the simulation. Use your

imagination. Do not worry about whether it is possible to build the ticket machine. What is required is a machine that is really easy to use.

To measure your success, ask yourself this question: Does the machine enable users to buy tickets more quickly and easily than buying them from the ticket office?

Feedback on Activity 2.3 - Designing a New Ticket Machine

Clearly there is no right answer to this question. You could have designed a system that uses voice recognition so that the user simply says 'I'd like a return to central London', and the machine dispenses the correct ticket.

But is that sensible on a noisy station platform? Maybe you could have a machine with built-in mind reading capabilities, so all the user has to do is stand next to the machine and think 'I'd like a return to central London' and the machine dispenses the ticket. Remember we said that we did not have to actually implement the system.

Maybe you designed a machine with a map on it and the user points to where they want to go. This would get around the problem of the user wanting to go to a station near to Alexandra Palace, but not knowing exactly which one. But there are hundreds of stations in a city like London. It would have to be a big map.

If you read Prof. Thimbleby's paper (http://www.uclic.ucl.ac.uk/projects/tvm/), you will see his solution: to do away with ticket machines altogether. You would have a travel card or an electronic device that allows you to be identified: your mobile phone maybe. The system knows when and where you got on the train, where you get off, works out how much money you owe for the journey and debits your account. No need for tickets at all so no need to queue in stations and no need to decide in advance where you want to go.

Another point that Thimbleby makes is the need to look at the larger, complex system in which an interactive system (in this case, the ticket machine) is embedded. Here, that is the existing railway ticketing system, including its regulations, inter-organisational interfaces, accounting and legal framework, etc.

If your redesign was like the simulation, but you simply made the buttons bigger or more colourful, or made the instructions clearer, then you missed the point of the exercise. The original system had some fundamental problems. Try again, and don't be afraid to think dramatic changes, so long as those changes make life easier for the user.

Chapter 3

Activity 3.1 - Generating Questions for the Five Modules of Simplex One

Follow this link: http://www.uclic.ucl.ac.uk/projects/tvm/simulation/index.html

This is a simulation of a train ticket-booking machine as used in the south of England. Press the 'Station context' and 'Front panel' buttons to see photos of the machine on a station platform and a close up of the front of the machine.

The simulation is based on the main interactive part of the machine where the user selects their destination and ticket type (the part of the machine with a big '1' pointing to it on the front panel photo).

Uncheck the box on the right marked 'Prefer photos' to remove the tick (so that you don't get photos). You may also want to enlarge the window as much as possible.

Press 'Cancel' to start the simulation working. The arrow buttons down the left and right hand sides act like the buttons on the actual machine, and the central frame shows the display. The idea is to press the arrow that points to the operation you want to perform. So if you want to buy a ticket to Cambridge, for example, you should press the fourth button from the top on the left.

Have a few minutes' play with the simulation to familiarise yourself with it.

For this activity:

- Generate one question and suitable responses using the five modules of Simplex One.
- Evaluate the train ticket-booking machine based on these questions.

Feedback on Activity 3.1 - Generating Questions for the Five Modules of Simplex One

Five Design Principles

Zone One: Perception
Question: "Does the system provide suitable visual or auditory input to the people using it?"

Responses: very suitable - suitable - unsuitable - very unsuitable

Zone Two: Responses
Question: "Does the system require reasonable responses from users?"

Responses: very difficult - difficult - easy - very easy

Zone Three: Working memory
Question: "Do the people who use the system have to keep too much information in mind when using it?"
Responses: very easy to keep in mind - easy to keep in mind - difficult - very difficult

Zone Four: Long Term Memory
Question: "Does the system require you to hold too much knowledge in long-term memory?

Responses: Much too much knowledge required - too much required - acceptable level required - easy level of knowledge

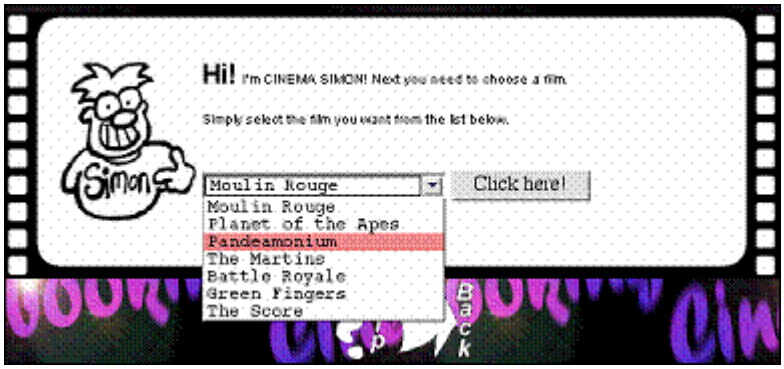
Zone Five: Executive Functions
Question: "Does this design require tasks that are too difficult?"

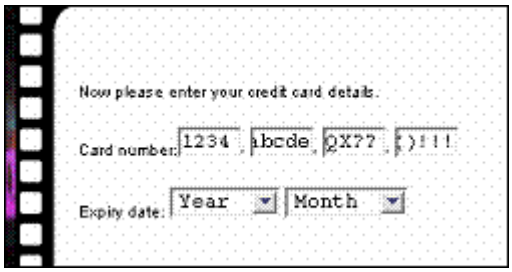
Responses: too difficult - difficult - - easy - very easy

Activity 3.2 - Heuristic Evaluation

Apply two of Nielsen's heuristics - 'User control and freedom' and 'Error prevention' to a Cinebooking website.

Two screen shots of this system are given below:





Feedback on Activity 3.2 - Heuristic Evaluation

User Control and Freedom
The system should allow the users to recover from mistakes efficiently and painlessly. The users should have facilities that allow them to control how they navigate through the system. The system should not impose its own navigation on the users.

Good point
On every page, there is a back button and a help button, which are prominent and clearly labelled. If the user gets confused, they can easily 'exit' to the help system. If they make a mistake (e.g. select a film they don't want to see) they can apparently go backwards in the dialogue to correct that mistake. Because these options are available on all pages, the user has control over their navigation through the site.

Bad point
However, the button labelled 'Back' takes the user back to the beginning of the dialogue, not back one step as one would expect. Therefore, the system imposes extra work on the user if they press the 'Back' button. They will probably have to work their way through several pages to get back to where they want to be. Hence, the 'Back' button, while on the surface supporting user control and freedom, actually does not because it does not behave in the expected way.

Error Prevention
Users will make mistakes. It is best to identify what those mistakes may be and design the system to prevent them occurring, rather than allowing them to occur and tidying up the mess afterwards.

Good point
The drop-down menus used to the select the cinemas, times and films are good at preventing errors (see screen shot above).

This is because the user can only (for example) select a film that is showing. If instead of the drop-down menu, the user were given a free text entry box to type in the name of the film they wished to see, then they may make errors. They may, for example, type in the name of a film that is not available or misspell one that is available. The drop-down menu prevents these sorts of errors from occurring.

Bad point
In contrast, the text entry box for credit card details allows users to type in any combination of letters, numbers or symbols.

A credit card number must be four groups of four numbers, and the system could prevent the user typing in any text that does not fit that pattern. Because in this case the system allows the user to type in erroneous data, this part of the system does not conform to Nielsen's definition of error prevention.

Chapter 4

Activity 4.3 - Using UCSD

In this activity, we want you to try out (briefly) the UCSD approach to designing an application. Please note that the emphasis is on the methods. We do not expect you to produce a fully featured application.

Your task is to create a notebook in which you note your thoughts and ideas on a daily or weekly basis:

- Use the UCSD approach
- You want to be able to search the online version for particular words or phrases. You think that the ability to make links (hyperlinks perhaps?) between different entries in your notebook might be very useful as you progress through the book. You imagine that this might also be very helpful for organising your thoughts.
- Do you need to get feedback from users? For the moment, this is intended as a personal application, for your own use only. But, who knows, perhaps you could package this notebook and sell it commercially? Unless you decided it Open Source it, of course.
- Construct a simple working prototype so that you can evaluate your design.

Feedback on Activity 4.3 - Using UCSD

Remember that the UCSD approach is iterative. In this Activity you will probably work through the UCSD Requirements Gathering - Design and Storyboarding - Prototyping - Evaluation loop several times.

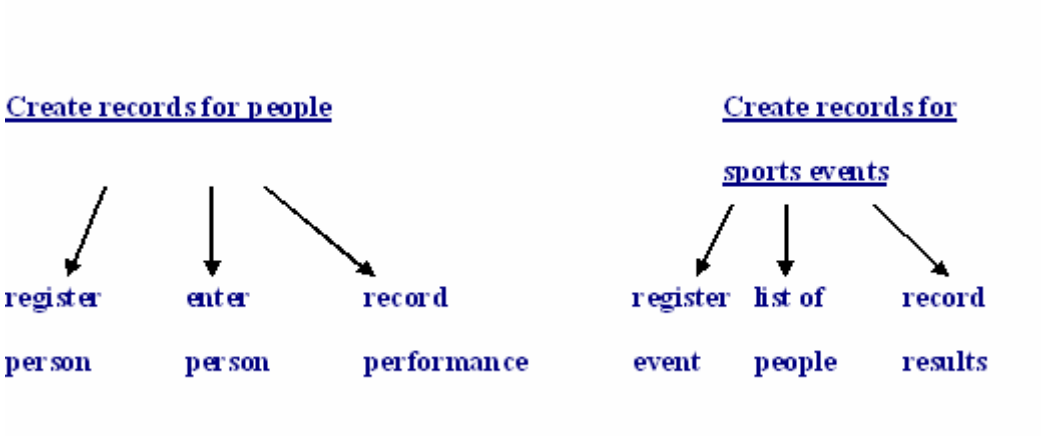
If you asked your fellow students for feedback on your design, what usability heuristics did they use?

Are some of your fellow students more constructive in their evaluations than others? In this book, we show you how to undertake practical systems development and encourage you to develop critical reasoning skills based on a sound appreciation of theoretical principles.

Chapter 5

Activity 5.1 - Practice at Task Analysis

Here is a simple task analysis of a system to organise a sports day at a school. Please consider the task analysis, evaluate it and suggest any improvements.



Feedback on Activity 5.1 - Practice At Task Analysis

Logically, this analysis seems fine. You have two classes of objects, namely people and events. You have actions for each. The problem is that this approach generates duplicate data. If you consider 'records for people', this contains the full set of data. The same data are repeated under 'records for sports events'.

You could do without either. The problem is that if your records are based upon people, it will be difficult to ask questions about events. For example, what was the fastest time in the egg and spoon race? Conversely, if you base your system on sports events, questions about people might pose a problem. For example, how did Fred do in the marathon?

Perhaps the task could be depicted in a different way? If so, what do you suggest?

Activity 5.2 - Different Task Analyses

You are organising a book sharing club and you want to design a simple computer-based system to keep records. There are ten members of the book club. The basic idea is that every member buys up to ten books a year. Before they buy a book, each member must find one other member who agrees to the purchase. This is so as to encourage the purchase of books of wider interest to the club. The member who buys the book retains ownership of it. Any member can request to borrow a book from another member for up to four weeks.

You want to have a system that records book purchases, who bought each book, who supported its purchase, who is currently borrowing it and when they are due to return it.

Your first step is to set up a card system with a card for every member and for every book.

Conduct a task analysis of this task.

How would this task analysis help you with subsequent stages of development if you were using the UCD process?

One of your (respected) colleagues is concerned about errors in a computer system and suggests that any new data should be input twice so as to avoid errors. (There have been fierce rows in the past about ownership and possession of books.) How would this design feature change the system and would it be a good idea to include it in the task?

Feedback on Activity 5.2 - Different Task Analyses

To create a hierarchical task analysis for this system, you need to realise that you are dealing with two classes of objects: people and books. You then need to carry out activities on the information about each person and each book. For example, for person 'AB' propose to buy book XY etc. For book YZ, being read by 'AC' etc.

When you consider building a computer-based version of this system, would you need / want to enter information twice?

If you take this approach, then you may well have to enter every item of information twice: once for the person and once for the event. If you wish to enter data only once, you could have automatic transfer of data from books to people or vice versa. Alternatively, you could organise your system by people only. You would need to have a method of dealing with queries about books!

Activity 5.3 - Different Task Representation

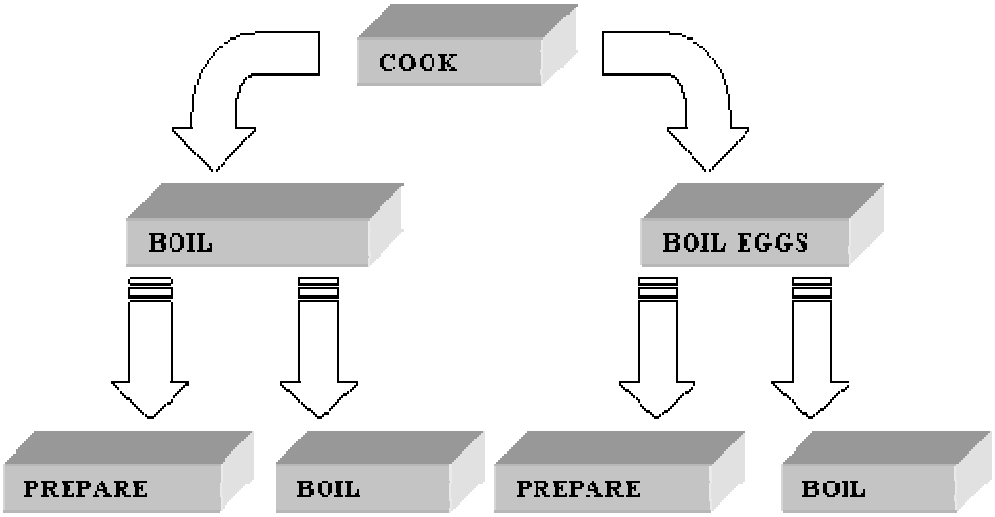
It is important to realise that a hierarchical task analysis depends upon the task involved having at least a reasonable resemblance to a hierarchical structure. What I want you to do now is to design two tasks. The first will have a hierarchical structure. The second will NOT have a hierarchical structure. You may consider the two examples that I gave you of a hierarchical task and a matrix-based task, but please do not copy either of them. Design your own tasks and portray them in your own way. Please try to be as original as possible in your selection, design and portrayal of the two tasks. Make them as different as possible from the examples provided.

When you have designed your two tasks, write a brief discussion of how you would conduct an HTA for each of them. Is it possible to conduct an HTA for both tasks? What are the merits and problems of doing an HTA on your first task? What are the merits and problems of doing an HTA on your second task?

Feedback on Activity 5.3 - Different Task Representations

You may see distinct differences between your two tasks. What are those differences - and what are their implications for subsequent task analyses? A hierarchy allows different tasks to be broken down into different sub-tasks. A matrix or grid expects cross-references to occur over the rows and columns. Can a hierarchy depict a task based on a matrix? The answer, surprisingly, is yes, but the hierarchy effectively ignores the occurrence of the same subtasks in different tasks. They are set out as separate tasks.

For example, the following task can be depicted as either a hierarchy or a matrix:



You will see how the lowest level tasks are repeated in both halves, but this is not acknowledged by the structure in any way. The questions remain: what is the most natural and accurate way to represent the tasks and are they the most useful? At a conceptual level, which is the most accurate representation of how we think? If the users use matrix-like memory structures, perhaps a matrix would provide a better task analysis? If the users use hierarchical memory structures, would that provide a better task analysis?

Activity 5.4 - Conducting a Task Analysis

Now use your two tasks to conduct an HTA on each, even though the second task is not hierarchical in design. Set out the textual description of the subtasks and the sets of plans to guide their use. Evaluate your two HTAs, identify their strengths and weaknesses, then decide which is the better (if any) and why.

Feedback on Activity 5.4 - Conducting a Task Analysis

You are probably getting better now at conducting task analyses. You may have found that it is possible to do an HTA on both tasks. (That is why HTA is so popular.) However, you will almost certainly find it more difficult to do a matrix analysis on both tasks. If you do not believe me, try it now!

Hierarchies can depict most structures. In your view, is it better to depict a task by a hierarchy or by its most natural form? Do you have a natural preference for one type of structure over another? Should we match the form of the task analysis to the forms preferred by users?

Activity 5.5- Evaluating Task Analyses

Building upon activity two and three, now carry out a SWOT analysis of your two TAs. Follow the four headings defined below:

SWOT (Strengths, Weaknesses, Opportunities or Threats)

Strengths: What are the good points of this TA?
Weaknesses: What are the weak points of this TA?
Opportunities: How would this TA provide a good basis for system design?
Threats: What problems would you encounter in using this TA as a basis for system design?

Now compare the two TAs. Decide which, if either, is better.

Feedback on Activity 5.5 - Evaluating Task Analyses

Try to be as impartial as you can in this activity. Try not to be protective of your work (not easy, as many designers demonstrate).

All else being equal, you may have found the first task analysis (based on a hierarchy) to be easier to do. The second, (based on some other structure) may have been more difficult. Try to distinguish problems due to the structure of the task from those due to other factors. For example, one of your tasks may have been a more complicated structure. Perhaps the practice on the first task made the second one easier to do!

Decide, if you can, which of the TAs was the better and why. Finally, explore briefly some of the other methods of task analysis mentioned in this chapter.

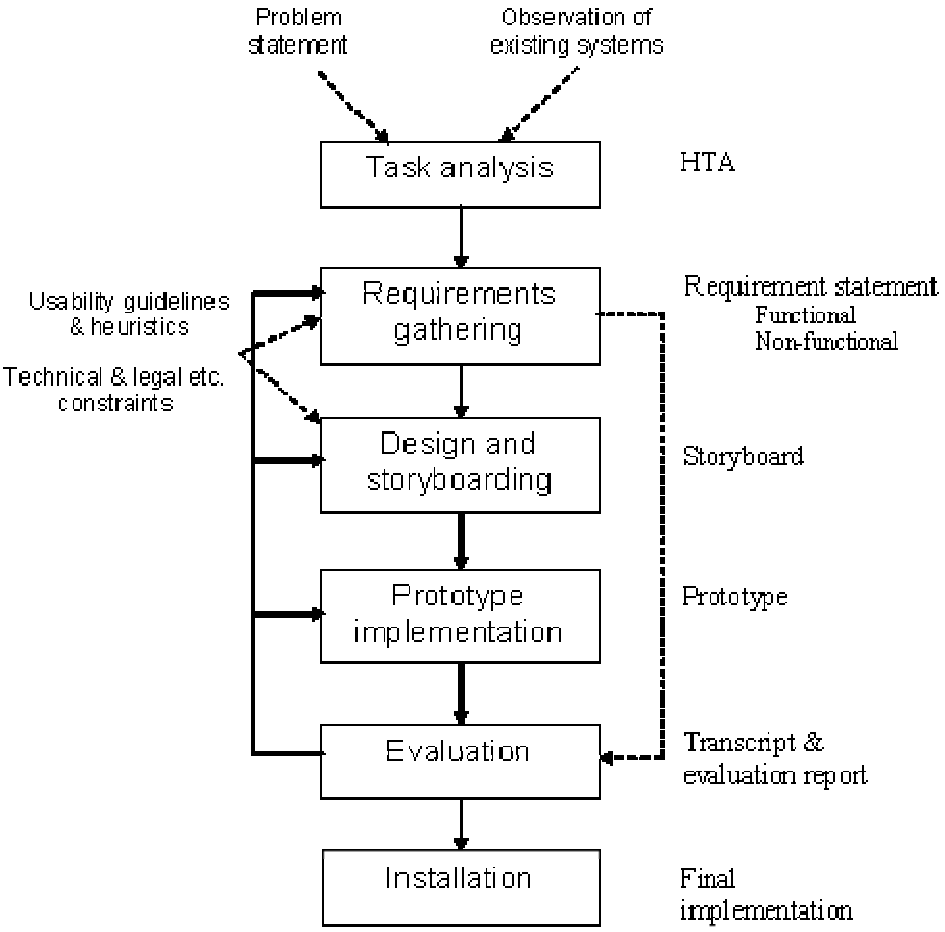
Activity 5.6 - User Centred System Design And Task Analysis

Explain how task analysis contributes to the stages of the UCD process.

Feedback on Activity 5.6 - User Centred System Design And Task Analysis

Task analysis is the first main stage of UCD. It will influence all subsequent stages. Requirements gathering, for example, will take as its input the output of task analysis. The danger of task analysis is that it may focus too much on the task itself and pay less than sufficient attention to users' psychological resources and preferences.

Design will be influenced by both task analysis and requirements gathering. In some cases, the task analysis may need to be revisited if the design stage proves to be too difficult to realise. Perhaps more clarification is required or a more modest set of requirements accepted. The creation of a storyboard scenario will reflect or challenge the output of the task analysis. System evaluation will be informed, in part, by the task analysis. If problems are identified at this stage or during maintenance, the task analysis may help to clarify what can be done about it.



Chapter 6

No online activity

Chapter 7

Activity 7.1 - Evaluating an Existing System

Take a well-known word-processing application such as MS Word and use it to produce a brief document. When you were working, were there any commands or functions that you could not remember properly? Note any such events. Did you use the help function? How did you overcome the memory problem? If you have not done so before, try to use MS Word to insert a hyperlink, to split a screen or to arrange a list in ascending and then descending order.

Feedback on Activity 7.1 - Evaluating an Existing System

The typical applications designer must rely on the users having reasonable powers of memory. They cannot be always explaining where to find functions, there would be no room on the screen. Some functions are so well buried that they cannot be found even with the "help" of the help function. I have tried to suggest functions that you might have not used before. However, try to identify functions that you use rarely at all. Make a note of them (but not how to use them) and several days from now, try to remember how to use them. I once

spent two hours looking for a function that I needed to use. All applications of any size, present memory challenges to users.

Activity 7.2 - URLs and Memory

List a dozen (12) URLs of different websites with which you are familiar. They should be of different lengths. Write them out in any order. Study them for a minute and then put the list away. Go away and do something else for an hour. Then come back and write out the list of URLs again from memory.

Feedback on Activity 7.2 - URLs and Memory

This activity demonstrates the fallibility of human memory. You may have been able to remember only a few URLs. Don't worry: that is the normal result. Initially, individuals who browse the web mainly use search engines. As people become more familiar with what they want, they make less use of search engines. Instead, they go directly to the site they want.

The importance of easy to remember websites and web addresses is increasing. Make sure that users find it easy to remember where to find you. Addresses should be as short and as meaningful as possible. Avoid unusual features unless they are memorable.

Activity 7.3 - Perception and Information

Without referring to the list of URLs you prepared for Activity 7.2 , try to find one or more of the websites from memory. Use Google to assist you if necessary. Can you remember what the homepage of one of the websites looks like? Can you describe it or draw a rough sketch of it?

Feedback on Activity 7.3 - Perception and Information

Did you have problems remembering how to find the website? The longer you leave it, the more intervening tasks you do, the more difficult it becomes. This is a common experience, so try to make it easier for your users!

Unless you found a particularly simple website or have a photographic memory, you will have overlooked many features on the website. This may because they were either unimportant to you or because they were not obvious or salient. Human perception is highly selective, often in quite complex ways, so remember to direct your users in your designs.

Chapter 8

No online activity

Chapter 9

No online activity

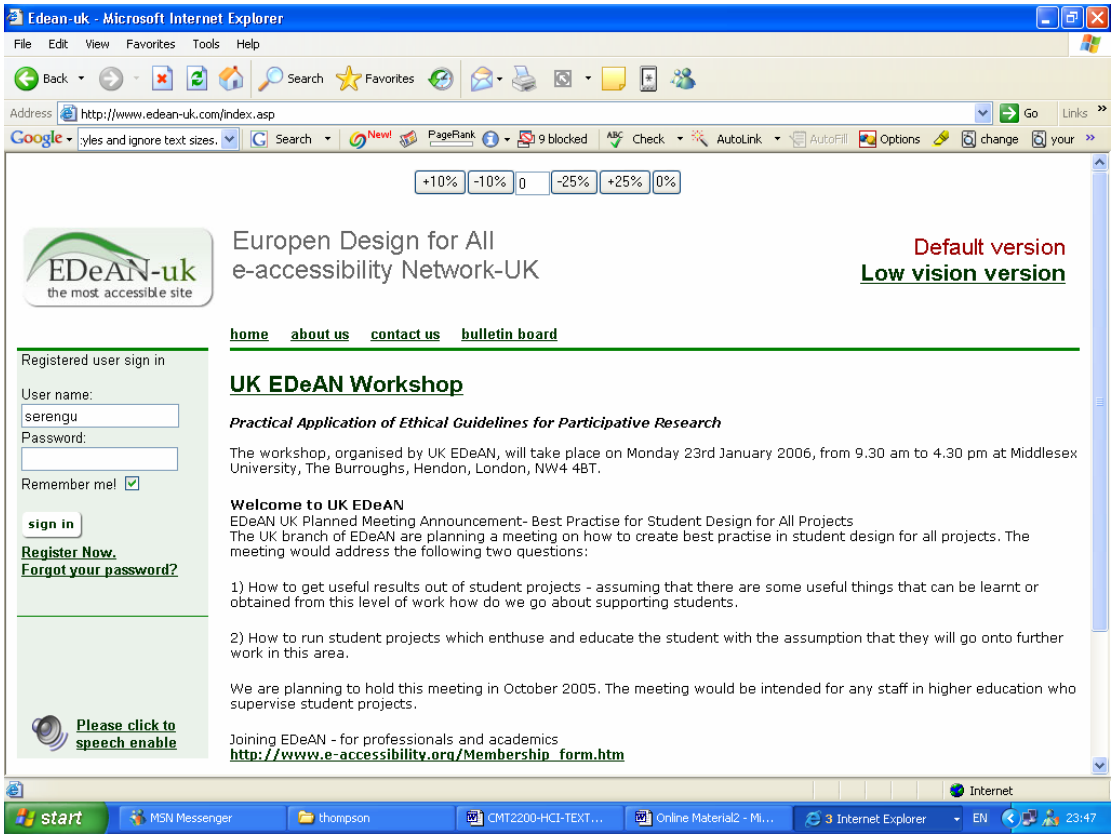
Chapter 10

No online activity

Chapter 11

Activity 11.1 - Test Your Website's Accessibility

In this activity, you should assess the accessibility of edean-uk website (http://www.edean-uk.com) as shown below or any website you wish to test.



Alternatively, you can test Mac OS X Panther v10.3 which can be found at http://www.apple.com/macosx/features/universalaccess/

Feedback on Activity 11.1 - Test Your Website's Accessibility

Check first to see if your web authoring tool or the program developer's website can check for valid code and accessibility. Ensuring that HTML and any coding such as CSS is correct aids accessibility (http://www.w3.org)

Try any or all of the following:

- change your browser colours, for example, white font on black background and then change Internet options (from the Tools menu), followed by checking boxes in Accessibility options, eg ignore page colours, ignore font styles and ignore text sizes.
- change your screen resolution to 1024 x 786 | 800 x 600 | 640 x 480. Many partially sighted people use a 640 x 480 screen size (to be able to this go to your computer's control panel then select display option and then finally select Settings)
- change your computer display settings options, eg to large, high contrast white or yellow on black
- disable frames if your site uses them
- try greyscale or print the pages in black and white to see if the contrast between colours is sufficient to distinguish elements
- if your browser allows it, try using your own stylesheet

During this test you may ask the following questions to yourself

1. Does the site still make sense and does it still function as planned?
2. Is the structure consistent and therefore predictable?
3. Is the navigation intuitive?
4. Has any information been lost?
5. Is there good colour contrast?
6. Are all parts of the site accessible?
7. Do all visual elements have appropriate alt text and full descriptions if necessary?

Chapter 12
No online activities