

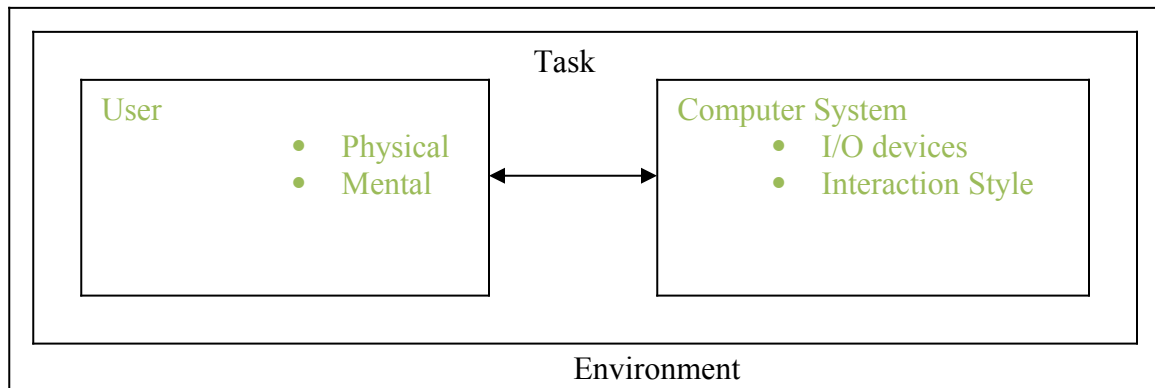
HCI -14- TASK ANALYSIS

Aims and Objectives: in the previous lecture we looked at some of the design issues involved in HCI and discussed a number of techniques that may be used. This lecture focuses on the importance of understanding the user's task in carrying out HCI design. In particular it describes a technique called Hierarchical Task Analysis (HTA) and demonstrates how it can be used.

Lecture Notes

Introduction to Task analysis

We can start by revising a diagram included in the first HCI lecture. It showed the user interacting with a computer system to accomplish some task within an environment. In previous lectures have looked at the user and at the computer system. In this lecture we will focus on analysing specific tasks that the user may engage in.



Task analysis is not just one technique but the name for a whole group of techniques that focus on understanding how the user goes about their tasks. The particular technique we will look at is called hierarchical task analysis (HTA) and is based on decomposing task into smaller sub-tasks and so on in a hierarchical manner.

In many ways task analysis is similar to requirements elicitation as carried out during traditional systems analysis. The emphasis however is slightly different. Task analysis is about understanding the task (including non-computerised aspects) rather than analysing the functional requirements of a system.

Task analysis can be used in a number of ways. Some of these are summarised in the table below.

What the user is Analysing / Doing	Purpose of Task Analysis
Performing a manual task	Computerise or provide computerised support for the task
Performing a task using an existing computerised system	Rewrite or update the system
Performing a task using an existing computerised system	Evaluate the usability and fitness for the task of the system
Performing a task using an existing computerised system	Provide documentation or training for using the system

Before looking at hierarchical task analysis in detail we look at a refinement to the Interaction Loop that was covered in an earlier lecture. The refinement provides a richer framework for understanding human-computer interaction and includes consideration of the task.

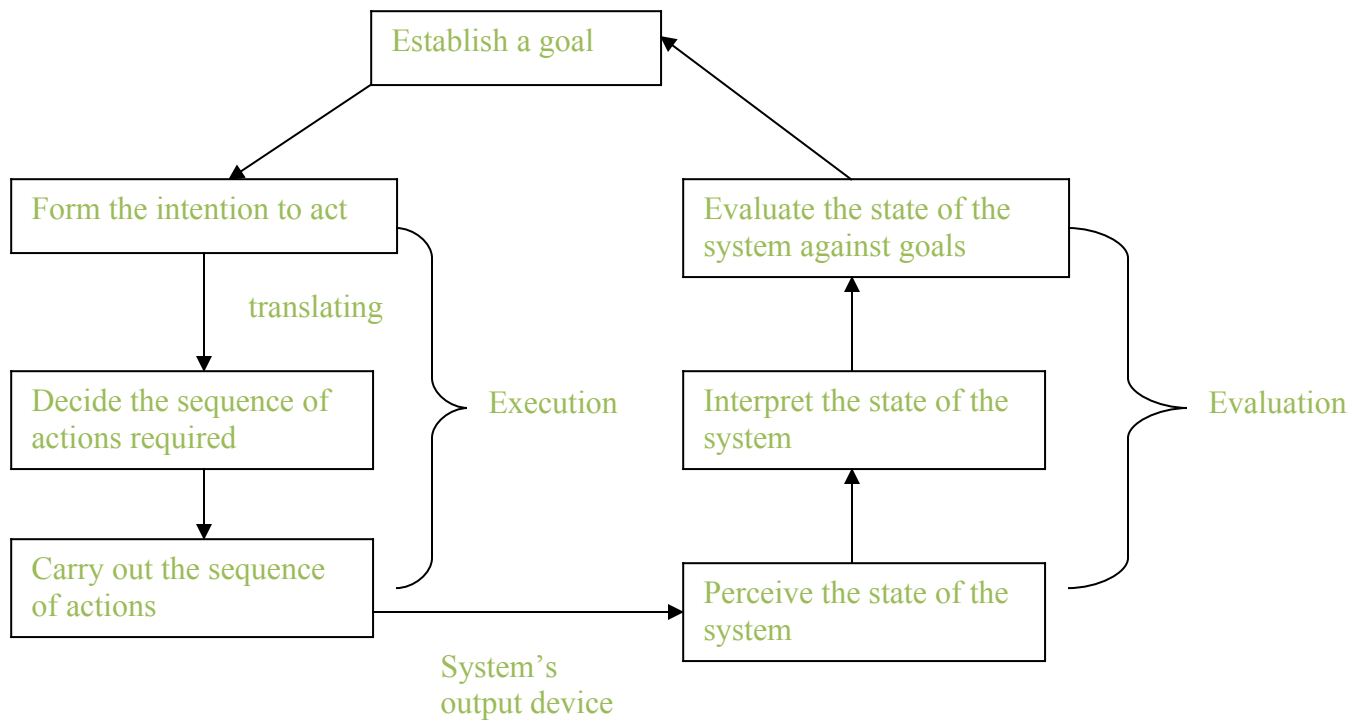
Norman's Execution Evaluation Cycle

In the lecture about interaction Styles we looked at the concept of the Interaction Loop, which is a very abstract framework for thinking about how a user and a computer system interact. It consists of four stages that may be carried out repeatedly in a loop. The four stages are:

- User interacts with system via its input devices
- System's state is changed by the user's input
- System's output devices reflect the system's new state
- User perceives the new state of the system

What is missing from this framework is my consideration of the role of the task. But if you think about it people don't normally engage in random interactions with a computer system. There is always some purpose. We always use a computer to **do something** e.g. write a letter, play a game, search for information on the Internet, transfer money from one account to another, land a plane etc. etc.

Donald Norman is the author of a hugely influential book called "The Design of Everyday Things." He has developed a model of Human – Computer Interaction that does take account of the role of the task. It is usually referred to as *Norman's execution-evaluation cycle*. The cycle consists of the seven phases shown below.



The idea is that the user starts by formulating a **goal**. The **task** is to perform the necessary actions to achieve the goal. Example goals might be to write a letter, send an email, get a high score in a game of ‘Mash the mauve-eyed monsters of Thrall’, create a graphic to be used as an advert on the corporate web site etc, etc. In order to achieve the goal the user **formulates the intention to act** and decides the **sequence of actions** they need to perform (e.g., start up MS Word, copy an existing letter to the same person, change the letter text, print the letter and save the file). Depending on factors such as the **complexity** and **familiarity** of the task this may or not require conscious thought. The user will then **carry out the sequence of actions**. Carrying out the actions will change the state of the system in some way and this will be conveyed to the user via the system’s output devices. The user will **perceive** the system’s state **interpret** it. For example they may realise having printed the letter that they had forgotten to change the date on the letter to today’s date. The user will compare their understanding of the system’s state with the state it should be in when they have achieved their goal. If, as in the example given, the user realises that they haven’t yet achieved their goal they will formulate a new goal (e.g. correct the date on the letter) and the cycle will start again.

The Gulfs of Execution and Evaluation

According to Norman, the ease or difficulty with which the user can achieve their task using the system is influenced by two factors which he terms the Gulf of Execution and the Gulf of Evaluation. Depending on the operations that the system allows and the way they are presented, the user may have to do more or less work to translate their intentions into a sequence of actions that will achieve their aim. *The width of the Gulf of Execution is a measure of the difficulty that the user may face in making this translation.* If it is too wide then it is likely that they will make errors. *For example, some people experience an uncrossable Gulf of Execution when faced with programming their video recorder.* They know what they want to record but can't work out the exact sequence of actions needed.

The size of the Gulf of Evaluation is a measure of how difficult it is for the user to work out the current state of the system and whether their actions have been successful in achieving their goal. For example, new computer users often find it difficult to work out where the software they are using has saved their documents. They sometimes perceive that the system has lost their work when in fact it is just that they can't find it. This gap between what the system has actually done and what the user perceives it as having done is an example of the Gulf of Evaluation.

Norman's model of the execution and evaluation cycle is a powerful one that can be used for analysing the interaction between user and computer system. The concepts of the gulfs of execution and evaluation are particularly useful in understanding why some systems are easier to use than others. In the next section we go to look at a detailed method of analysing the way in which users carry out tasks.

Quick Quiz questions 1

What are the seven stages in Norman's execution evaluation cycle?

Hierarchical Task Analysis

Hierarchical Task Analysis (HTA) analyses tasks by decomposing them into sub-tasks. The result is a hierarchical structure plus a procedural description of how the sub-tasks are carried out e.g. their sequence, whether they are conditional etc. The procedural descriptions are called **plans**.

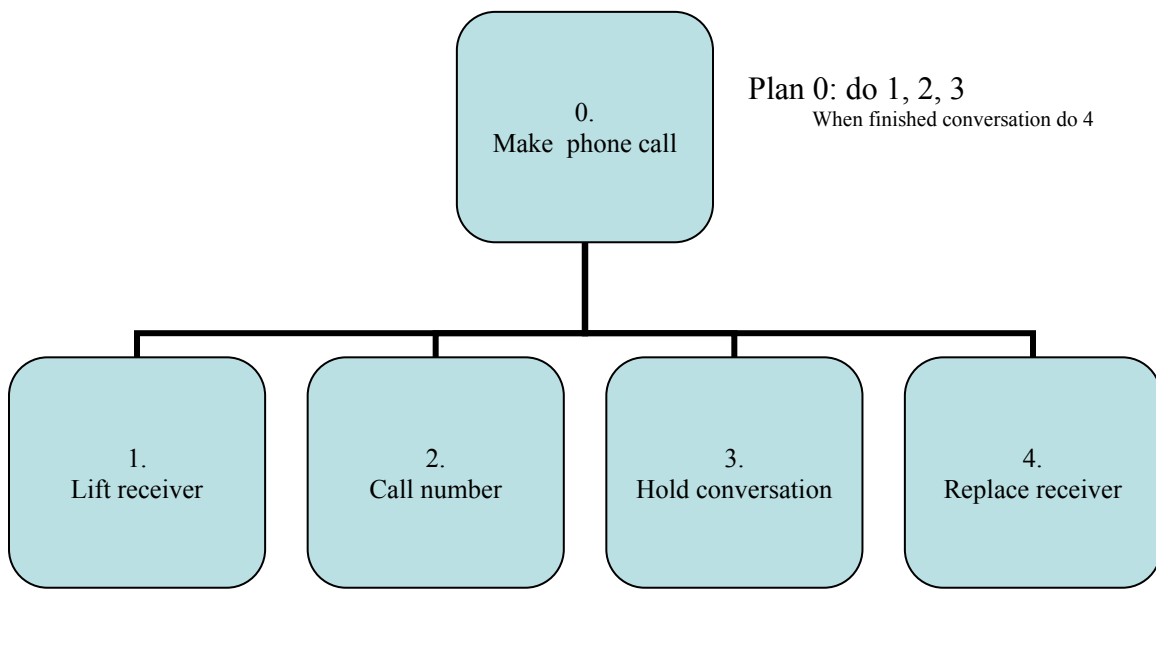
The best way to understand HTA is by looking at an example. Realistic examples tend to be rather complex so we will look at a simple example of carrying out task analysis on the everyday task of making a phone call.

The overall task is to make a phone call. At its simplest this could be decomposed into the subtasks of: lift receiver, call number, hold conversation and replace receiver. This can be documented as shown below (textual diagram).

- O. Make phone call
 - 1. Lift receiver
 - 2. Call number
 - 3. Hold conversation
 - 4. Replace receiver
- Plan 0: do 1, 2, 3,
- When finished conversation do 4

Notice that the task and subtasks are numbered and that a plan has been added. The numbering of the plan shows that it describes how to carry out task 0 (i.e. the overall task). The plan indicates that subtasks 1, 2 and 3 are carried out in sequences and that subtask 4 is carried out conditionally on the conversation finishing.

A graphical notation can also be used to document the results of HTA. The diagram below gives the same information as shown above.



The lines drawn under each of the subtasks indicate that they cannot be decomposed any further.

The degree to which you decompose the task depends on the purpose for which you are doing task analysis. It is a matter of judgement so there is no absolute or wrong answer. Let's assume that in this example it is significant to take into account whether the number is dialled in full or is pre-programmed into the phone and is recalled using memory keys. We would enhance our model as shown below.

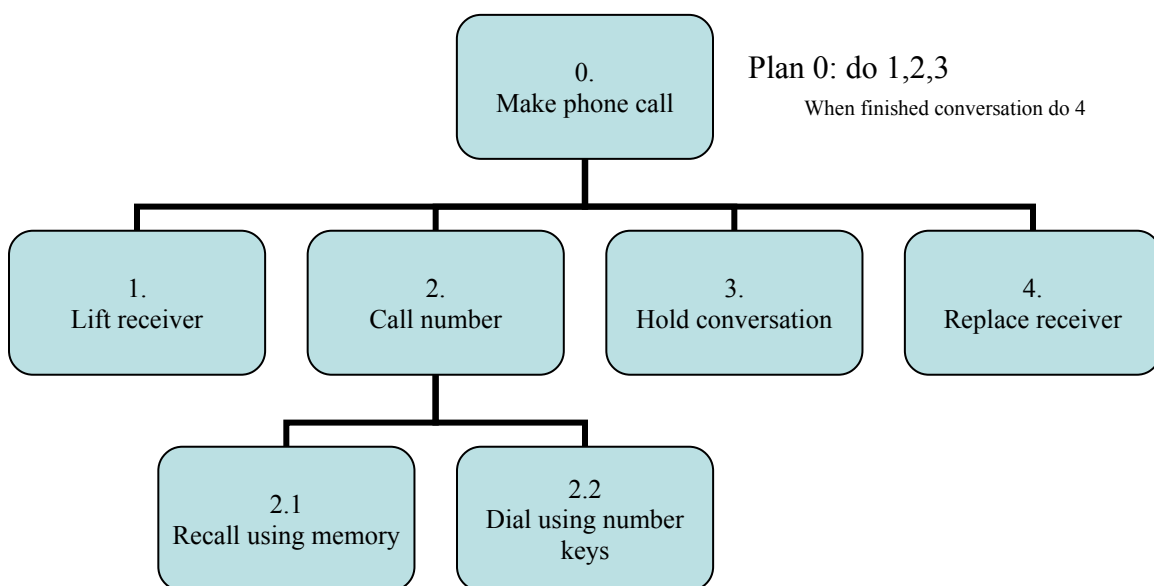
0 Make phone call

1. Lift receiver
2. Call number
 - 2.1 Recall using memory key
 - 2.2 dial using number keys
3. Hold conversation
4. Replace receiver

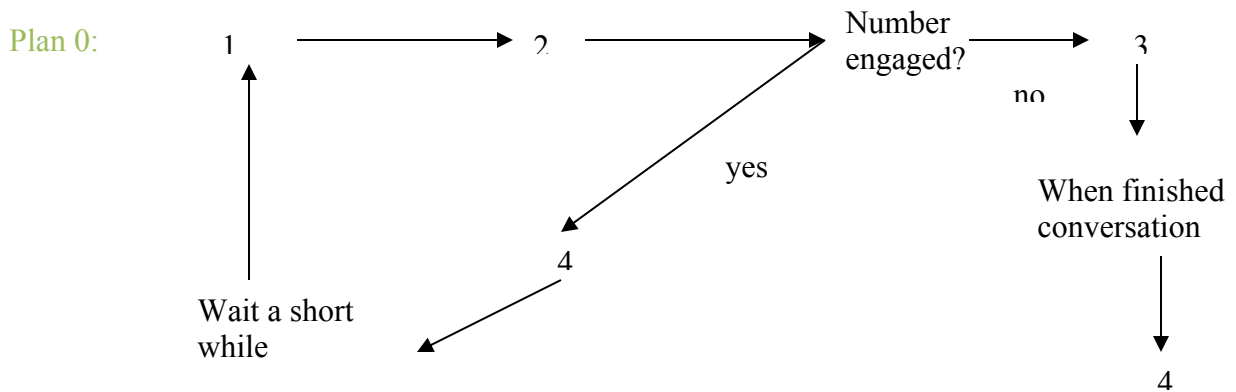
Plan 0: do 1, 2, 3
 When finished conversation do 4

Plan 1: do 2.1 if the number is programmed into the handset
 Otherwise do 2.2

Task 2 has been decomposed into two subtasks. A plan has been added to indicate that either subtask 2.1 or subtask 2.2 is carried out depending on whether the number is pre-programmed into the phone or not. The graphical version of the hierarchy is given below.



Now, let us assume that we need to take into consideration the situation where the number dialled is engaged. Let's say that if the number is engaged the user replaces the receiver, waits a short while and then tries the number again. We could indicate this by enhancing plan O as shown below.



Note that I have chosen to use a short flow-chart based diagramming technique for showing the repetition in plan 0. This is a matter of choice and I could equally well have used a pseudo-code style of documentation. It depends on what seems clearest and most appropriate.

Quick Quiz Question 2

Here is a simplified description of how I would print a document that I am currently editing from PC. Firstly, I check to see if the printer is switched on and switch it on if it isn't on already. Then I click the print icon. If the printer runs out of paper while it is printing my document I add more paper to its paper tray,

Carry out a HTA on the description and document the results in both graphical and textual form. Keep the analysis very simple, for example don't worry about whether the printer is plugged in to the power or not.

Gathering Data for Task Analysis

The techniques used to gather data for task analysis are similar to those used in traditional systems analysis. The main methods are outlined briefly below.

System **documentation** such as procedure manuals may be a good place to start. It must be remembered however that documentation describes how things should be or were expected to be done but not necessarily how they actually are done.

Observation of users carrying out tasks is one of the most valuable techniques. The observation may be carried out under controlled conditions or in the user's normal environment. Often it is necessary to record the session some way e.g. video tape. A common technique is to get the user talk through their actions (either as they perform the task or afterwards) to explain their reasons for doing things.

Interviews can be used as very direct way to get information from users and experts. They may be used in conjunction with observation in order to get more in depth of detailed information. Unstructured interviews are often used early on in the process so that the analyst can get to know the user and find out background information. Structured interviews with pre-set questions may be used later on in the process to gather more detailed and precise information.

The use of observation and interviews are covered in a little more detail in the lecture on evaluation techniques.

Using the Results of Task Analysis

The results of task analysis can be used in a variety of ways and contexts. Two particular example uses are discussed below.

Task analysis is often used in order to aid the design or **user documentation**. User documentation may include reference materials and tutorials. It may be in the form of online documentation (e.g. online help or online training) or it may be paper based. Task oriented documentation is often more appropriate than documentation organised around the developer's view of the system structure. Task oriented documentation may be particularly useful to novices who may not want or need an in-depth explanation of all the options that a system provides but may benefit from a series of tutorial that walk through common tasks.

Another use of task analysis is to inform the **detailed design of elements of a system interface**. Sequences of operation that are frequently carried out together should obviously be made easy for the user. This may influence the placement of operations e.g. those operations that can be accessed directly through icons and those used less frequently that may be hidden away as menu options. Wizards may be designed to lead users through a series of actions that are frequently carried out in sequence. In this case the results of the HTA may translate directly into the logic encoded into the wizard.

Quick quiz Questions 3

Identify three techniques that may be used to gather information for task analysis

Summary:

This lecture looked at task analysis. The desire or need to perform a task can be seen as the force that drives the Interaction Loop covered in the earlier lecture on Interaction Styles. Norman's execution-evaluation cycle is a useful framework for analysing how users perform tasks. It highlights two possible problem points, namely, the Gulf of execution and the Gulf of Evaluation.

Hierarchical Task analysis (HTA) is a technique for analysing tasks in detail. It decomposes tasks into subtasks and identified the order in which tasks are carried out using 'plans'. The results of HTA can be documented diagrammatically or in a textual layout. The source material for HTA can be gathered using the traditional systems analysis methods of looking at documentation, observation and interviews. The results of HTA can be used in a number of ways e.g. in the creation of user documentation or to help design detailed aspects of the system interface (e.g. menus).

Having completed this lecture you should:

- Discuss the role of task analysis in HCI
- Use Norman's execution –evaluation cycle to perform a very basic analysis of human computer interaction.
- Carry out task analysis of a simple human computer interaction and document it using HAT

Tutorial Exercises

These tutorial exercises ask you to carry a simple HTA for the task of using a search engine on the web to find something. It is very easy to get buried in detail when doing task analysis so try to keep things simple. Remember there is not absolute right or wrong answer. The purpose of the exercises is just to give a flavour of what it is like to carry out HTA.

1. Observe someone (it can be yourself if you are not in position to work with anyone else) searching for information on some topic of interest using a popular web search engine. Make notes about the process. What are the main steps? What order are those steps carried out in? Are some steps optional? Are some steps repeated?
2. Try to document the task you observed using HTA. Try both a diagrammatic and textual representation. You may find it easiest to take a very simplistic approach at first and then add refinements. Try not to get lost in detail during this stage.
3. Use the results you produced for exercise two to draw up an outline of a tutorial to teach someone how to use the search engine.