The use of Task Analysis in HCI

The listing of actions a user carries out in performing a task.
For example, a person preparing an overhead projector for use would be seen to carry out the following actions :-

1. Plug in to main and switch on supply.
2. Locate on/off switch on projector
3. Discover which way to press the switch
4. Press the switch for power
5. Put on the slide and orientate correctly
6. Align the projector on the screen
7. Focus the slide

In Human Computer Interaction, task analysis is the recording of physical and perceptual actions of the user whilst executing the task.

---

2. What are the aims of Task Analysis?

- Task analysis verifies that the set of actions employed by the user does accomplish the task.
  Task analysis explicitly describes the procedure that the user actually employs since this may be different from the expected series of actions.

- Task analysis is used to:
  predict the time taken to learn a new task and become a proficient user of the particular application / machine. Task analysis may reveal how difficult one method is to learn compared to another.

  predict the time taken for a proficient user to accomplish the set task - this can reflect whether the interface is good at supporting exploration. Is it quicker to simply explore by trial and error or attempt to find out through help?.

  predict the time taken for expert execution of the set task - how long does it take to become expert once a procedure has been discovered? This can be affected by the design of an interface.

  The latter point is an important application of task analysis. In 'Phoneline' banking or telephone directory enquiries, for example, the time spent with each customer can be reduced by milliseconds as a result of task analysis. Here speed, accuracy and efficiency are vital and task analysis comparison studies help determine a more effective interface design.

- To improve the delivery of information to the user.
  This involves identifying any problems with the delivery of information to the user and the consideration of possible solutions. These issues will be dealt with in later lectures.

3. What is the range of user actions that Task Analysis can examine?

Task analysis may consider one or more of the following user actions:
- Machine input using the keyboard and/or mouse .
- Other physical actions, such as the need to lift the mouse to rezero, head movements required when copy typing
- Perceptual or cognitive actions such as knowing that the mouse pointer must be moved onto the appropriate title of menu bar and clicked to see the contents of the pull-down menu
- Mental actions involved in decision making or memory recall, for example selecting the appropriate icon from the tool bar

4. What are the user tasks that can be analysed?

- Task analysis might focus strictly on the procedure for getting the task done. Here the rate of learning is not an issue whereas the correct and/or speedy operation of the procedure is of paramount importance. A procedure could be very difficult to learn but there might be a requirement of **no residual errors** by a trained user.

- The converse may also be the case where the analysis is focused on the discovery stage. The focus might be on how to do something new and/or to investigate if the interface supports exploration - trial and error versus on-line help.
- Task analysis may also consider the cost of learning a procedure. Once a new procedure is discovered , how long does it take to become expert? This can be influenced by the interface design. It is important not to reinforce the older slower procedure.
- Analysis might focus on the procedures for error recovery. Certain errors need elaborate recovery procedures, e.g. vi editor. There was not a recovery procedure for the error of **declaring a man dead!**
  "There are more ways of making errors than doing it right!"

  User procedures for error recovery must be simple. It is perhaps a neglected area by designers and analysts

Task Action Mappings

- When analysing actions used in a certain task, it is useful to think of the user making a **directional link from** the task **to** the action. Such linkings or mappings are called **Task Action Mappings**.
- Ideally, the mapping should be linear, **one-to-one** i.e. each task leads to the user to think of one action.
- If the mapping is convergent, **many-to-one** i.e. a number of distinct tasks lead the user to think of one action, then there may be no problem with such inconsistency. For example, the two tasks of renaming a file and moving a file in Unix may both be associated with the one action of using the mv command.
- If the mapping is divergent, **one-to-many** i.e. one task is linked to more than one action, inconsistencies may arise.
  For example, the perceived task of deleting files or directories in Unix may be associated with two actions, using the commands rm and rmdir.
- Given that a user thinks in terms of Task Action Mappings, then an interface should reflect this in the way it gives the user information.
  For example, a manual listing what each command does i.e. action leading to task, may be much less helpful to someone learning a system than a manual which lists the tasks which can be done and gives the commands to use i.e. task leading to action.