Task analysis

Summary

Task analysis analyses what a user is required to do in terms of actions and/or cognitive processes to achieve a **task**. A detailed **task analysis** can be conducted to understand the current system and the information flows within it. These information flows are important to the maintenance of the existing system and must be incorporated or substituted in any new system. **Task analysis** makes it possible to design and allocate tasks appropriately within the new system. The functions to be included within the system and the user interface can then be accurately specified.

Benefits

Provides knowledge of the tasks that the user wishes to perform. Thus it is a reference against which the value of the system functions and features can be tested.

But note that according to the USERfit guide:

> The reader should be aware that **task analysis** can be a very time consuming activity if used with a high degree of detail on complex problems. ... It is possible to get caught in what is loosely termed '**analysis** paralysis', where more and more detail is investigated.

Method

Task decomposition

The aim of 'high level **task** decomposition' is to decompose the high level tasks and break them down into their constituent subtasks and operations. This will show an overall structure of the main user tasks. At a lower level it may be desirable to show the **task** flows, decision processes and even screen layouts (see **task** flow

analysis, below)

The process of **task** decomposition is best represented as a structure chart (similar to that used in Hierarchical **Task Analysis**). This shows the sequencing of activities by ordering them from left to right. In order to break down a **task**, the question should be asked 'how is this **task** done?'. If a sub-**task** is identified at a lower level, it is possible to build up the structure by asking 'why is this done?'. The **task** decomposition can be carried out using the following stages:

1. Identify the **task** to be analysed.

2. Break this down into between 4 and 8 subtasks. These subtasks should be specified in terms of objectives and, between them, should cover the whole area of interest.

3. Draw the subtasks as a layered diagram ensuring that it is complete.

4. Decide upon the level of detail into which to decompose. Making a conscious decision at this stage will ensure that all the subtask decompositions are treated consistently. It may be decided that the decomposition should continue until flows are more easily represented as a **task** flow diagram.

5. Continue the decomposition process, ensuring that the decompositions and numbering are consistent. It is usually helpful to produce a written account as well as the decomposition diagram.

6. Present the **analysis** to someone else who has not been involved in the decomposition but who knows the tasks well enough to check for consistency.

Task flow diagrams

Task flow **analysis** will document the details of specific tasks. It can include details of interactions between the user and the current system, or other individuals, and any problems related to them. Copies of screens from the current system may also be taken to provide details of interactive tasks. **Task** flows will not only show the specific details of current work processes but may also highlight areas where **task** processes are poorly understood, are carried out differently by different staff, or are inconsistent with the higher level **task** structure.

More information

USERfit: **Task analysis**
A detailed description of how to carry out **task analysis**

Crystal, A & Ellington, B (2004). **Task analysis** and human-computer interaction: approaches, techniques, and levels of **analysis** Proceedings of the Tenth Americas Conference on Information Systems, New York, August 2004.
A comparison of hierarchical and cognitive **task analysis**.

Hackos, J. & Redish, J. (1998). User and **Task Analysis** for Interface Design. Chichester: Wiley.

Crandall, B., Klein, G., Hoffman, R. R. (2006). Working Minds: A Practitioner's Guide to Cognitive **Task Analysis**. MIT Press.
"This is an important book for the engineering of complex systems and information technology systems."

Tools

TaskArchitect is a tool that supports Heirarchical **Task Analysis**.

Variations

If the tasks are already well understood, it may be sufficient to just identify and document the tasks as part of context **analysis**.

Next steps

If other requirements activities are complete, move on to design.

Background reading

Shepherd, A. (1985). Hierarchical **task analysis** and training decisions. Programmed Learning and Educational Technology, 22, 162-176.
Shepherd, A. (1989). **Analysis** and training in information technology tasks. In D. Diaper, Ed. **Task Analysis** for Human-Computer Interaction, pp.15-55. Chichester: Ellis Horwood.

Nielsen, J (1994) Extending **Task Analysis** to Predict Things People May Want to Do