Chapter 8. Task Analysis

Table of Contents

Context

In the last unit we looked at user models and modelling. Users usually have goals and carry out tasks to achieve those goals (interacting with a computer system as necessary). In this unit we will look at ways of modelling tasks. There are different types of task model. All of the techniques concern goal-oriented things that people do but some focus on the actions carried out to achieve a goal; some on the knowledge required to carry out a task; and, others on the relationships between the objects involved in the task. This unit will review three major types of model (Hierarchical Task Analysis, Knowledge Based Analysis and Entity Relationship modelling). You will see how they are employed in systems development. We will focus on HTA and you will learn how to produce this sort of analysis and how the results can be used.

Objectives

At the end of this unit you will be able to:

- Explain the purpose of task modeling
- Distinguish between task analysis and other analyses such as GOMs (Unit 7)
- Distinguish between HTA, KB and ER analyses.
- Give a brief review of KB and ER analyses and explain how they can be used.
- Carry out HTA for a range of user goal types.
- Explain and show how results of HTA can be used to improve interaction.

Unit Plan

In previous units you have focused on design (units 2, 3 & 6) and on users (units 4, 5 & 7). But of course users and computers do not exist in isolation: apart from some games applications, computers are there to support users in going about their every-day work. To be useful, computer systems have to help users to do the things they need to do. To be usable, they need to help the users do them in a way that seems natural to those users. Task analysis is concerned with understanding how users go about their work at the moment, so that new computer systems can be designed to help with that work rather than hindering it or forcing it into a different pattern.

In this unit, we will study three different approaches to analysing users' tasks:

- **Hierarchical Task Analysis (HTA)** involves describing users' tasks in terms of the activities involved at different levels of detail. Superficially, it looks a bit like the GOMS analysis that you studied in unit 7 but, as you will see later in this unit, it is concerned with larger-scale tasks in the work domain, rather than user tasks (involving the details of thinking and acting with a particular device).
- **Knowledge-Based (KB) Analysis** involves creating taxonomies of objects that are important within the work domain.
- **Entity-Relationship (ER) Modelling** involves describing the entities (objects and actors) involved in the domain and relationships between those entities. You may have come across this type of approach before, if you have studied systems analysis, but don't worry if you haven't.

In this unit, we will cover how to create each of these types of representation of a task and how to use that representation to reason about design. We will focus most on HTA.

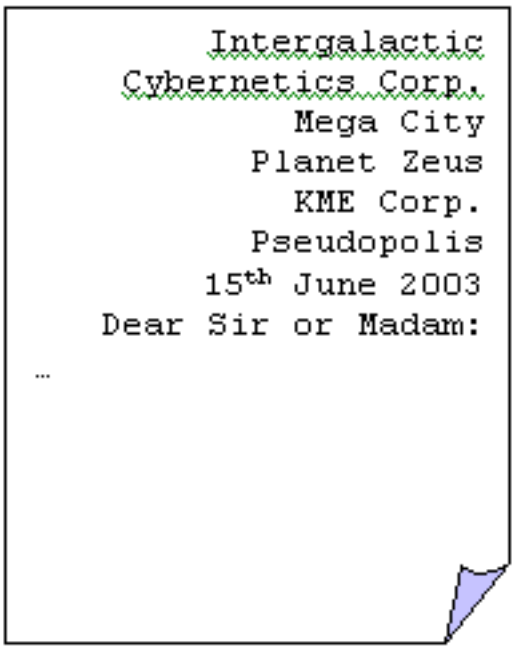
Introduction to this Topic

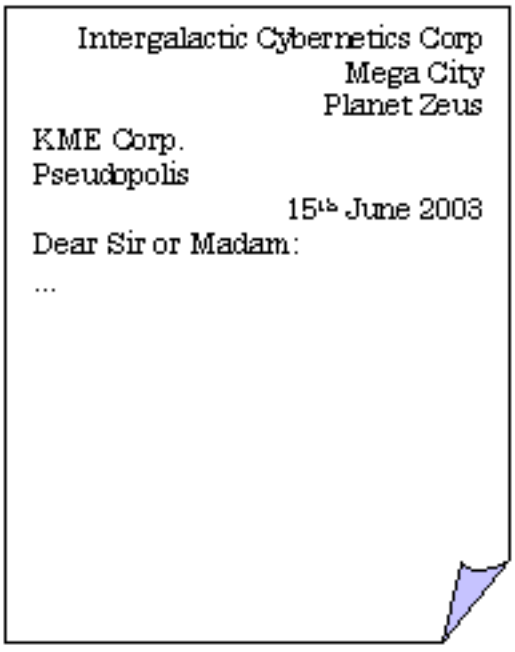
W'll start with a simple example, based on a simple task of writing a letter and preparing it for posting.

First, we observe Linden, who has no computer support, performing this task in her office. She finds a sheet of paper, a matching envelope and a pen, and settles down at her desk to write. She starts by putting her own address in the top right-hand corner, then picks up her address book (which is lying on the desk beside her) and flicks to the page containing the addressee's address information. She copies the addressee's address to the paper (left-hand side), then writes today's date and starts on the body of the text:Dear...". When she has finished, she signs the letter then reaches for the envelope, writes the name on it and copies the address from the address book onto the envelope, folds the letter, puts it into the envelope and seals the envelope. Task completed.

We then go to the next office to observe Remi using his word processor to do something similar. He settles down in front of his computer and opens his word processing package, using a document template that already includes his own address and today' s date. He opens his electronic address book, finds the appropriate address, and uses his mouse to copy and paste the address from the address book into his document. He starts on the body of the text:Dear...". When he has finished composing the letter on the screen, he selects the print option and waits until a message appears on his screen to say Your document Letter to KME has been printed on Laser 6.He gets up and walks out to the shared printer area, heads for the printer in the far corner, and flicks through the pages that have accumulated there until he finds his letter. He picks it up, walks back to his desk, reads it through and signs it. The printer does not print on envelopes, so he finds a sheet of mailing labels that will fit through the printer. He has a document template that is set up to print in the format appropriate for mailing labels, so he opens that and copies the information from his electronic address book to this new file. He selects the print option, remembering to select the option for manual feed (so that he can place the blank mailing label sheet in the manual feed tray of the printer). He then walks out to the printer and inserts the mailing labels in the manual feed tray then hurries back to his office to click on the OK button that has now appeared next to the message manual feed: click OK when ready to print. He walks back to the printer again to collect the printed label, then goes to find an envelope. He folds the letter and slips it into the envelope, sticks the mailing label on the envelope, then seals it. Task completed.

Which of these procedures is better? It depends on what the aims of the letter-writing exercises are. The word-processed version will look more professional, whereas the hand-written one will look more personal. The word-processed letter may be easier to read (depending on how legible Linden's handwriting is). Which was produced more efficiently? That probably depends on how quickly Linden writes, and how quickly Remi types. For short letters, it is quite likely that the hand-written one will have been produced more quickly. Computerisation does not always improve efficiency.





Is task analysis about reproducing exactly the same situation with a new design as the one we are replacing? Absolutely not! Intelligent design is about adapting and improving the work system. So we see in this small example that early on there were things that could be done more efficiently with the word processor than by hand: by integrating address book information with the letter-writing tool, the act of transcribing could be made more efficient. You can also imagine that if this letter were a fairly standard one, Remi could have cut and pasted text from another document to speed up the writing of this one. However, later stages (which involved integrating tasks with the word processor with printing tasks) were, frankly, tedious: although we feel that it ought to be easier to print a mailing label from pre-existing information, in practice the system used by Remi does not support this.

Domain tasks and device tasks

In the discussion of letter-writing, we saw that very similar end objectives could be achieved in very different ways using different tools (pen or word processor plus printer). The two scenarios described similar domain tasks – what the user is achieving in the world – but involved different device tasks – what the user had to do using the available tools to achieve that effect. One of the most important challenges of design is often to ensure that there is a good 'fit' between the domain tasks and the device tasks. In an ideal world, the device would be almost invisible to the user – not in the sense that it cannot be seen, but in the sense that it is not noticed.

We can draw on an analogy of a hammer: if you are hammering in a nail, things always go best when you are unaware of the hammer itself; it is just an extension of your hand and the hammering action is fluid and easy. Then you bang your finger, and suddenly that flow of activity is broken; you are intensely aware of the hammer as a separate object, and you have to work hard to restore the earlier productive hammering action. Tools should be ready-to-hand in the way that a hammer is to a carpenter: the user should not be fighting with the tool, but using the tool easily and naturally to achieve their tasks, almost unaware of its existence. For Linden, the worst things that might happen are her pen leaking or running out of ink. Remi's work is more prone to the tools getting in the way of the task, although, conversely, his tools may also provide him with more active support for those same tasks.

Of course, there are cases where the computer system is the focus of the activity: a good computer game is engrossing, but it does depend on the computer system being there and visible: it is the focus of attention. However, even in this case, the user should be able to focus on the game rather than being distracted by pictures going jittery, the sound fading unpredictably or other unintended features.

Of course, games are not the usual focus of task analysis, which is more commonly concerned with the productive kinds of tasks that are found in work environments (offices, factories, etc.). In the remainder of this unit, we will focus our attention on the kinds of work tasks that have a clear goal – typically an end product.

Data Acquisition

Task analysis of any kind starts from data – information about the work and the work setting. For HTA the data that is needed is data about procedures – about how activities are structured into tasks. For ER and KB analysis, the data that is needed concerns objects in the work domain and their interrelationships, and maybe also the actors (people that are essential to getting the work done).

In the scenarios above, we talked about going into Linden's and Remi's offices and observing them. Implicitly, while observing them we were also taking notes about their activities that could be used later for analysis. Observation and note-taking (or even video-recording) are one way of gathering the data that is needed for task analysis. Another way of getting such information is by interviewing users (i.e. the people who perform the tasks being studied). Again, responses can be noted by hand, or recorded using a tape-recorder. Each of these techniques has advantages and disadvantages.

Interviewing is often more efficient than observation. As people talk about their work, they use a particular vocabulary that expresses their understanding of what they do and how they work that can usefully be adopted in any computer support for their tasks. You find out how they think they perform their jobs and the things that they perceive as being important.

Unfortunately, people are often unable to articulate what they really do. This may be because they are so immersed in their working environments, so familiar with the way things work, that they take much of their knowledge for granted. Often the most important things about the way they work are the ones that are most obvious to them, and therefore the ones they will forget to mention. Also, as they become experts at their tasks, they forget all the details: they develop ‘compiled skills’ which becomes difficult for them to break down and describe fully. To take a simple example that may be familiar to you: if you are driving a manual car, as you pull away from a junction you will change up the gears. When do you change gear (e.g. from third to fourth)? What is the detailed procedure you follow as you change? Do you take your eyes off the road for any reason? Learner drivers are very aware of the procedure; each foot movement, each glance in the mirror, the act of feeling for the gear lever (maybe looking for it) are all deliberate actions. As expertise develops, the conscious awareness of the process diminishes. To understand users’ tasks in detail, it is often necessary to observe them, to identify taken for granted knowledge and compiled skills.

A third source of data for task analysis is often existing documentation: there may be procedure documents that describe how tasks should be performed (which may or may not describe how they are actually performed). In what follows, we are assuming that data has be acquired from somewhere, and we focus on ways of analysing that data.

Design Evolution and Revolution

Before we start, though, a note of caution about the limits of task analysis: it’s no good for revolutionary design. Every now and then, a totally new kind of design appears on the scene – one that really changes the way we do our work, or think about interaction. Two of the examples from recent years that you will be familiar with are the Graphical User Interface (GUI), which moved us forward from the text-based command-line interfaces of an earlier generation and made computing much more accessible to non-specialists, and the World Wide Web, which has revolutionised information access and hastened the globalisation of interactions and commercial transactions. For every high-impact success like these, there are untold numbers of failures – revolutionary ideas that simply failed to catch on and faded into oblivion.

Contrast these with the word processor, which evolved from the typewriter and retained many features of its predecessor: each generation of word processor has moved a little further from its origins, as users find new uses for the current generation of machine and adapt their behaviour to the new possibilities, which are then ‘designed in’ to the next generation products, so that we get co-adaptive behaviour (with users adapting to new systems, which are then re-designed to support new uses, which…)

Task analysis provides good support for design evolution, where understanding of the current task – the task structure, the entities and actors – is used as a starting point for new design. It is no use for design revolution, which typically involves creating new possibilities, that simply did not exist before, and introducing new concepts (e.g. the graphical object or the hypertext link).

Hierarchical Task Analysis (HTA)

Hierarchical Task Analysis has been in use for a long time – since the 1960’s or even earlier. It is most suitable for analysing tasks that have a well-defined structure – that is, tasks which tend to be performed in similar ways every time, rather than those that have a very loose structure. HTA involves describing the task in terms of a task-subtask hierarchy and a set of plans that define in what order subtasks may be performed, or under what circumstances particular subtasks are performed at all.

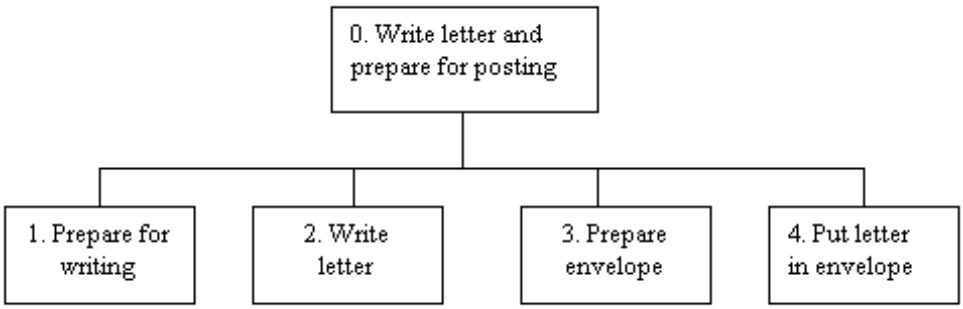
Example

We will start with the simple example introduced earlier: writing a letter and preparing it for posting. Initially we consider the description of Linden's activity:

Description of task	**Subtasks**
First, we observe Linden, who has no computer support, performing this task in her office. She	Prepare for writing.

Description of task	Subtasks
finds a sheet of paper, a matching envelope and a pen, and settles down at her desk to write.	
She starts by putting her own address in the top right-hand corner, then picks up her address book (which is lying on the desk beside her) and flicks to the page containing the addressee's address information. She copies the addressee's address to the paper (left-hand side), then writes today's date and starts on the body of the text: "Dear …". When she has finished, she signs the letter	Write letter
& then reaches for the envelope, writes the name on it and copies the address from the address book onto the envelope, &	Prepare envelope
& folds the letter, puts it into the envelope and seals the envelope. Task completed.	Put letter in envelope

One of the standard ways of presenting a HTA is as a tree structure:



There may be some other notations you are familiar with where the order of appearance of boxes in the tree indicates ordering (probably left to right). This is not the case for HTA: the only thing that matters is the hierarchy. As well as presenting the hierarchy, it is necessary to describe plans that define the possible ordering of activities. In this case, a suitable plan would be:

Plan 0: Do 1, then 2 and 3 in either order, then 4.

Although tree structures are visually appealing – well, more appealing than the alternatives, anyway – they can be tedious to draw without a suitable tool. Therefore an alternative text-based notation that relies on indentation is often used. We will use this textual notation to expand the task description for the letter-writing task.

- 0: Write letter and prepare for posting
  - 1: Prepare for writing
    - 1.1: Get paper
    - 1.2: Get envelope
    - 1.3: Get pen
    - 1.4: Get address book (not explicitly stated, but clearly necessary)
  - 2: Write letter
    - 2.1: Write own address

    - 2.2: Write addressee's address
    - 2.3: Write date and "Dear..."
    - 2.4: Write body text of letter
    - 2.5: Sign off
  - 3: Prepare envelope
    - 3.1: Write name on envelope
    - 3.2: Write address on envelope
  - 4: Put letter in envelope
    - 4.1: Fold letter
    - 4.2: Place letter into envelope
    - 4.3: Seal envelope

Again, we need plans to describe how to perform each subtask:

- Plan 1: Do 1.1, 1.2, 1.3 and 1.4 in any order
- Plan 2: Do 2.1 then 2.2 then 2.3 then 2.4 then 2.5
- Plan 3: Do 3.1 then 3.2
- Plan 4: Do 4.1 then 4.2 then 4.3.

Task analysis involves generating as general a description as possible. So, for example, we might want to generalise tasks 2.1, 2.2 and 2.3 to a new task: write head of letter. Similarly, we might notice that it is not necessary to have the envelope to hand until the time when it is to be prepare, or the paper to hand until the point where the user starts writing the letter, but we need the pen and address book for both, so we might break down task 1. If we do these things, we get a new structure:

- 0: Write letter and prepare for posting
  - 1: Get paper
  - 2: Get envelope
  - 3: Prepare for writing
    - 3.1: Get pen
    - 3.2: Get address book
  - 4: Write letter
    - 4.1: Write head of letter
      - 4.1.1: Write own address
      - 4.1.2: Write addressee’s address
      - 4.1.3: Write date and “Dear…”
    - 4.2: Write body text of letter

    - 4.3: Sign off
- 5: Prepare envelope
- 6: Put letter in envelope

Again, we need plans to describe how to perform each subtask:

- Plan 0: Do 1, 2, 3, 4 and 5, then 6. 3 must be done before 4 and 5; 1 must be done before 4; 2 must be done before 5.
- Plan 3: Do 3.1 and 3.2 in either order
- Plan 4: Do 4.1 then 4.2 then 4.3.
- Plan 4.1: Do 4.1.1 then 4.1.2 then 4.1.3
- Plan 5: Do 5.1 then 5.2
- Plan 6: Do 6.1 then 6.2 then 6.3.

We see that now different parts of the task analysis are presented at different levels of detail. This is often thought of as a Bad Thing, but in this case it allows us to describe the optionally and alternative orderings within the task more clearly. As with most aspects of design, there is no perfect solution just solutions that are better or worse for particular purposes.

Exercise 1 HTA for word processing letter

Look at the paragraph that describes Remi's word-processing task (reproduced below). Mark up this paragraph into activities that sensibly constitute a subtask, using the analysis already presented for Linden's activity as a guide. Construct a HTA description of this task.

Note

We then go to the next office to observe Remi using his word processor to do something similar. He settles down in front of his computer and opens his word processing package, using a document template that already includes his own address and today's date. He opens his electronic address book, finds the appropriate address, and uses his mouse to copy and paste the address from the address book into his document. He starts on the body of the text: "Dear …". When he has finished composing the letter on the screen, he selects the 'print' option and waits until a message appears on his screen to say "Your document Letter to KME has been printed on Laser 6". He gets up and walks out to the shared printer area, heads for the printer in the far corner, and flicks through the pages that have accumulated there until he finds his letter. He picks it up, walks back to his desk, reads it through and signs it. The printer does not print on envelopes, so he finds a sheet of mailing labels that will fit through the printer. He has a document template that is set up to print in the format appropriate for mailing labels, so he opens that and copies the information from his electronic address book to this new file. He selects the print option, remembering to select the option for 'manual feed' (so that he can place the blank mailing label sheet in the manual feed tray of the printer). He then walks out to the printer and inserts the mailing labels in the manual feed tray then hurries back to his office to click on the "OK" button that has now appeared next to the message "Manual feed: click OK when ready to print". He walks back to the printer again to collect the printed label, then goes to find an envelope. He folds the letter and slips it into the envelope, sticks the mailing label on the envelope, then seals it. Task completed.

Using the guidelines for adaptation, and the example of modifications presented above, modify your HTA so that it better matches the general task of using a word processor such as that being used by Remi

Answer at the end of the chapter.

Exercise 2 Re-designing computer-based procedure

You should now have HTAs for two similar tasks: writing a letter by hand and word-processing a letter. Hopefully, they share many of the high-level tasks in common, and differ mostly in the detail. Use these HTAs as a starting point for adapting the computer-based task so that it is easier for users.

Answer at the end of the chapter.

Activity 1 - HTA for finding information

Construct a HTA for finding information on the Word Wide Web, starting with data gathering. If possible, you should observe a few other people (fellow students or friends) performing the task, and note down everything they do. If this is not possible, perform the tasks yourself, and note down everything that you do. Use this data as a basis for constructing a HTA. The information seeking should cover two different kinds of information seeking tasks, as described here.

- There is a shareware text on Task Analysis, written by Clayton Lewis and John Rieman, available on the web. Use the Google search engine (http://www.google.com [http://www.google.com/]) to start the search, and observe how people behave from there. When the text has been found, you might like to read some of it!

- Imagine you are planning a trip to Sydney (Australia) or Boston (USA). You need to identify a cheap flight and also find a suitable hotel to stay in. Get all the information that you would need to be able to just book over the Web or telephone to make a reservation.

Use the HTA produced as a starting point for thinking about how the searching task can be made as easy as possible for users. Note that the design of search engines is a separate topic that is beyond the scope of this course!

A Discussion on this activity can be found at the end of the chapter.

HTA Conclusion

One of the hardest things to grasp about HTA is how to construct a ‘good’ one. In one respect, HTA is very easy: do some hierarchies and some plans. In another, that superficial simplicity hides a deeper challenge: what makes a ‘good’ HTA is its fitness for purpose, and assessing that is usually difficult. For this reason, it is important to construct HTA descriptions, and then to modify and rework them, thinking about alternative structures, and about what level of detail it is most useful to go to.

As noted above, HTA differs from GOMS in that it considers only physical activity (not cognitive tasks), and focuses on plans (rather than selection rules). Conversely, large-scale tasks can be described using HTA whereas only a dedicated person with too much times on their hands would complete a GOMS analysis for such large tasks (it would take weeks of effort to do it thoroughly). If using HTA as a guide for implementation or for detailed documentation, a fairly fine grain of detail (down to the individual action level) may be necessary; where it is being used to help get an understanding of the domain tasks, less detail may be appropriate.

Review Questions

Review Question 1

A HTA description consists of two main components. What are they?

Answer at the end of the chapter.

Review Question 2

HTA is a type of Cognitive Task Analysis. True or false?

Answer at the end of the chapter.

Review Question 3

What is the starting point for HTA?

Answer at the end of the chapter.

Review Question 4

Describe the process of constructing a HTA description.

Answer at the end of the chapter.

Review Question 5

Conduct a HTA for an imaginary fast-food delivery outlet (e.g. pizza delivery), from the point where the customer rings in to place an order to the point where the person who has made the delivery returns with the payment.

Answer at the end of the chapter.

Knowledge Based (KB) Analysis

Whereas HTA is concerned almost entirely with procedures, KB analysis focuses on the things (at least in a loose sense) in the task domain, namely objects and actions. KB analysis involves creating taxonomies, or hierarchies, of objects or actions. Such taxonomies can be used in interface design – for example, by ensuring that related objects or actions are grouped together sensibly, and that all important objects are represented at the interface.

Example

Again, we will start with the simple example introduced earlier: writing a letter and preparing it for posting. Initially we consider the description of Linden's activity:

Description of task	**Objects**	**Actions**

First, we observe Linden, who has no computer support, performing this task in her office. She finds a sheet of paper, a matching envelope and a pen, and settles down at her desk to write.	`Paper`
	
	`Envelope`
	
	`Pen`
		`Find`

She starts by putting her own address in the top right-hand corner, then picks up her address book (which is lying on the desk beside her) and flicks to the page containing the addressee's address information. She copies	`Address (sender)`
	
	`Address (addressee)`
	
	`Address book`
	
	`Page`
		`Pick up`
		
		`Open & flick`

Description of task	Objects	Actions
the addressee's address to the paper (left-hand side), then writes today's date and starts on the body of the text: "Dear …". When she has finished, she signs the letter	Date Body text Signature	 Write Sign
& then reaches for the envelope, writes the name on it and copies the address from the address book onto the envelope, &	Envelope Address (addressee)	Pick up Write
& folds the letter, puts it into the envelope and seals the envelope. Task completed.	Letter Envelope	Fold Insert Seal

Looking at this list, we can immediately see that there are two kinds of objects: the physical objects with which Linden is working, and the component parts of the letter (the bits of writing that turn a blank sheet of paper into a letter). Arguably, there are other distinctions – for example, between the address that is a kind of thing-in-the-world (as used, for example, by the postal service) and the address as a small portion of text that contains information about the corresponding thing-in-the-world. Conversely, maybe the actions 'find' and 'pick up' are essentially the same for this task: both are concerned with the user being in possession of the associated object.

A first attempt at a taxonomy might look like this. We are using the terms AND, OR and XOR to indicate whether an object can be or have all the sub-components at the same time. Here, AND means 'all of', OR means 'any number of', while XOR means 'exclusive or, exactly one'.

```
Letter-writing object XOR

   Physical object XOR

      Paper XOR

         Blank sheet
         Letter

      Pen
      Envelope
      Address book

         Page (lots!)

   Portion of text XOR

      Address AND

         Owner XOR

            Sender
            Addressee

         Location XOR

            Letter
            Envelope
```

```
    Date
    Body text
    Signature
```

In this particular case, there are no ‘OR’ objects: most parts of the taxonomy are XOR (e.g. the address owner is either the sender or the addressee, and we are not considering the possibility that these might be the same), while we are also saying that an address has both an owner AND a location (where it is written). There are some quirks – for example, that a sheet of paper is either blank or a letter; we might wish to add a third option (partially written). We will not develop this taxonomy further, but quickly present the corresponding taxonomy for actions:

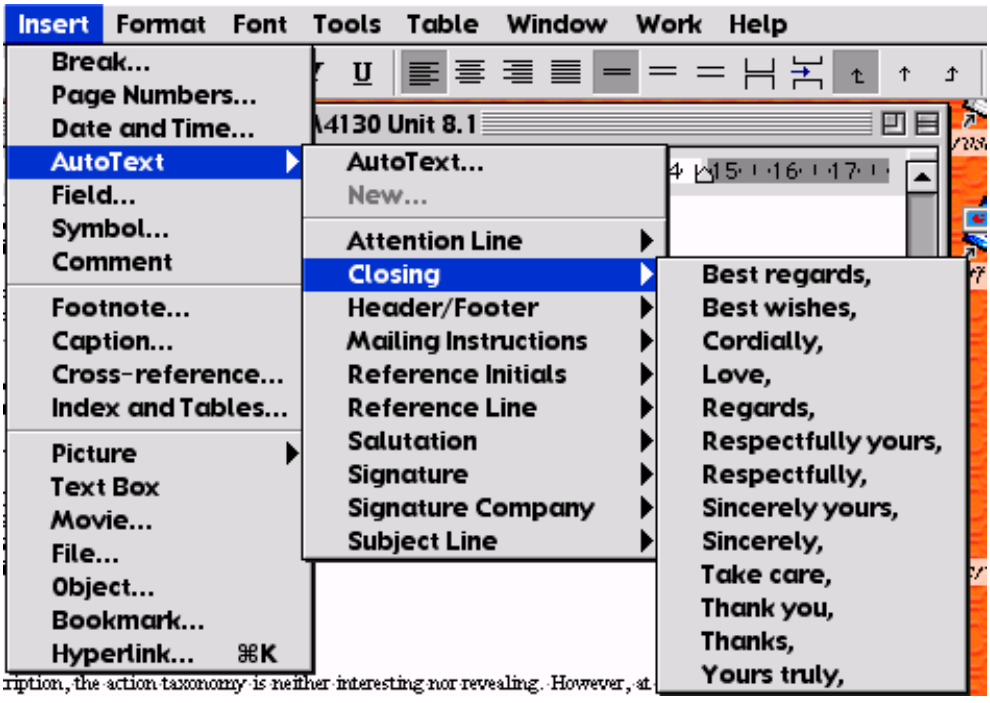
```
Letter-writing action XOR

   Gain possessionof
   Find (e.g. locate address in address book)
   Write XOR

      Write header information
      Write general text
      Sign

   Fold
   Insert
   Seal
```

At this level of description, the action taxonomy is neither interesting nor revealing. However, at a finer grain of detail it could be very valuable. Consider, for example, the pull-down menus implemented within word processors such as MS Word. If you look at the organisation of the menus under the top-level headings (‘file’, ‘edit’, ‘insert’, etc.), the groupings of items within each menu, and the options made available, you will see that there is a strong taxonomic basis to the organisation. See, for example, the Figure below. You can go one further and identify places where the taxonomic structure might be open to challenge (e.g. where should ‘insert table’ be located? And why does this ambiguity arise?).



As discussed for HTA, the level of detail required depends on the purpose. In this figure, we see a fine grain of detail (listing out all the most common closing phrases for letters), which is useful for this

level of implementation. In other cases, the challenge may be just to create a taxonomy of the physical objects in the domain to support reasoning about design alternatives, or to support the generation of overview documentation.

Exercise 3 KB analysis for the word-processor version of the letter writing task

Compare the KB description of Linden's task of handwriting a letter to the informal description (presented earlier) of Remi's task of word-processing one. Modify the object taxonomy presented for handwriting to describe the word-processing version of the task. Are there important design points that come out of this?

Answer at the end of the chapter.

Activity 2 taxonomic assessment of e-commerce sites

Select three e-commerce sites (e.g. a bookseller, a travel agent, a financial services supplier) and probe them to see whether you can identify any underlying knowledge based structure. What are the concepts the user has to work with? What actions are available to them? Are actions and objects grouped appropriately at the interface, or within the overall site?

A Discussion on this activity can be found at the end of the chapter.

Review Question 6

What are the core concepts expressed in a KB analysis?

Answer at the end of the chapter.

Review Question 7

KB analysis can be used to guide the design of manuals to support activity. True or false?

Answer at the end of the chapter.

Entity Relationship (ER) Analysis

Entity-relationship analysis is normally associated with systems analysis (notably database design) and, more recently, object-oriented programming, so if you have studied these topics, much of this section might seem familiar to you. Be warned, however: the task analysis we are considering here should have a much broader scope than ‘traditional’ ER modeling, being concerned with the entire work domain and not just the entities that will eventually be represented within the computer system.

Like KB analysis, ER analysis starts with the objects and actions involved in the domain. However, as well as the objects, we are concerned with their properties (and the distinction between objects and their properties, or attributes, is clearer in ER analysis than in KB analysis). Also, we are not concerned with the similarity (or otherwise) between objects, but with how they are related – for example, which actor in the domain performs particular actions on the objects.

Example

Yet again, we will start with the simple example introduced earlier: writing a letter and preparing it for posting. Since Linden's activity involves only one actor, we will only consider the more interesting case of Remi's activity using a word processor. We do something which might at first sight appear

strange, namely to include some of the machines as actors, when they behave in a way that is partially autonomous:

Description of task	Objects	Actors
… Remi settles down in front of his computer and opens his word processing package, using a document template that already includes his own address and today's date. He opens his electronic address book, finds the appropriate address, and uses his mouse to copy and paste the address from the address book into his document. He starts on the body of the text: "Dear …". When he has finished composing the letter on the screen, he selects the 'print' option and waits until a message appears on his screen to say "Your document Letter to KME has been printed on Laser 6".	Word processor Template Address Date Document Address book Body text	Remi. Printer Computer
He gets up and walks out to the shared printer area, heads for the printer in the far corner, and flicks through the pages that have accumulated there until he finds his letter. He picks it up, walks back to his desk, reads it through and signs it.	Printer Page Letter	Remi
The printer does not print on envelopes, so he finds a sheet of mailing labels that will fit through the printer. He has a document template that is set up to print in the format appropriate for mailing labels, so he opens that and copies the information from his electronic address book to this new file. He selects the print option, remembering to select the option for 'manual feed' (so that he can place the blank mailing label sheet in the manual feed tray of the printer). He then walks out to the printer and inserts the mailing labels in the manual feed tray then hurries back to his office to click on the "OK" button that has now appeared next to the message "Manual feed: click OK when ready to print".	Envelope Label Template Address book Manual feed tray OK button Message	Printer Remi Computer
He walks back to the printer again to collect the printed label,	Printer	Remi

Description of task	Objects	Actors

then goes to find an envelope. He folds the letter and slips it into the envelope, sticks the mailing label on the envelope, then seals it. Task completed.	`Remi`
	
	`Computer`	

The list of objects and actors presented so far may well be incomplete. In one respect this does not matter: task analysis is concerned with helping to get a good understanding of the task domain, and not necessarily with laying things out consistently and completely enough to support implementation. In another respect this is just a starting point: as the analysis proceeds, it becomes gradually more obvious what things matter and what don't, and we may want to expand detail in some places, while ignoring other issues.

There are already some obvious relationships – for example, that a printed page belongs to someone, that the computer displays a message to a person, that an envelope has a label (maybe), etc.. The process of KB task analysis involves identifying many more such relationships.

If we move on to consider actions, we note that actions are performed by an actor, who may use some instrument (or tool), and they change the state of some object. So, towards the end of the first stage of activity (as presented here), the printer is an actor that prints (action) the letter (object), whereas later the printer is an object into which Remi (actor) places the manual feed tray (object) containing mailing labels.

Exercise 4 ER analysis for the word-processor version of the letter writing task

Develop the ER description of the word-processing task using the ideas as presented in section7.5 of Dix et al. Are there important design points that come out of this? How does this description compare to the KB analysis you conducted earlier?

Answer at the end of the chapter.

Activity 3 entity - relationship assessment of e-commerce sites

Use the same three e-commerce sites as you used for Activity 2. This time, probe them to identify the underlying entities and relationships. Does this raise any new design issues that did not emerge during activity 2?

A Discussion on this activity can be found at the end of the chapter.

Review Question 8

What are the different kinds of entities that are generally listed in an ER analysis?

Answer at the end of the chapter.

Review Question 9

What is the most important difference between ER analysis as a form of task analysis and ER analysis for database design?

Answer at the end of the chapter.

Review Question 10

List some of the important relationships in ER analysis.

Answer at the end of the chapter.

Uses of Task Analysis

We have already summarised most of the important uses of task analysis:

- A source for generating documentation. By structuring the understanding of the task, it becomes much easier to structure a presentation of the task, in user-oriented documentation, whether that be structured around procedures, actions or concepts.
- A source for designing tutorial material. Like documentation, good, user-centred tutorial material that helps users learn to use a product is based around their tasks, so task analysis is a good starting point for designing effective tutorial material.
- Guiding system design. By focusing attention on the current system (and its strengths and weaknesses), task analysis can be used to design new interactions that have evolved in a reasonably natural way from existing practices, and to identify domain objects that need to be represented at the interface, and ways of grouping those objects.
- Requirements capture. Although task analysis refers to the existing system, rather than the planned one, it can help to structure requirements acquisition, particularly as users will often refer to the existing (familiar) system when discussing future requirements. In particular users may not find it easy to list features that should remain unchanged from the existing system, so the task analysis can help focus on what should stay the same as well as what should change.

To summarise: task analysis is necessary for bringing domain knowledge into the design, to make a new design or procedure as familiar and sensible and hence learnable as possible. Task analysis is not easy but then, neither is good design! It takes practice, and it also important to understand both the uses and the limitations of the various task analysis techniques available.

Discussion Topics

Use of HTA

How, if at all, could HTA be useful in designing e-commerce sites?

Comparing KB and ER analysis

Which, to you, seems the more useful technique: KB or ER analysis?

Answers and Discussions

Answer to Exercise 1

You should have a HTA description that describes hierarchies and plans for this task. You can compare this HTA with the handwriting one to identify the important ways in which the task has been changed when it was computerised. Which tasks have been changes substantially for example, made much longer or shorter, or with a different order of subtasks?

One of the purposes for HTA is to help design task structures that seem as natural as possible to users. One aspect of this might be to retain as much as possible that is already familiar to users; another might be to avoid unnecessary additional tasks.

Answer to Exercise 2

Adaptations are likely to include structural changes: for example, should each user have their own printer? Should printers have a special facility for printing labels, or even envelopes, without manual intervention?

Answer to Exercise 3

The word-processor taxonomy is likely to include labels as well as envelopes, owners of printed sheets as well as letters, and printers instead of pens. The multiplicity of printers with different names might be a design issue, as might the kinds of things that can be printed (printable items?). There are more design issues to be discovered!

Answer to Exercise

You should have listed some actors (a particular type of object) and actions they perform. You should have some events (such as 'when the letter has been printed'). You should have some action-event relations (e.g. the label must be put in the manual feed tray before it can be printed). For all the types of relationships, see if you can find at least one relationship in this word processing scenario.

By now, you have probably worked on this letter-writing task as much as you can face. Don't spend too long on this; what matters is that you are starting to get a feel for what is involved in conducting an ER analysis, and how it can help you understand more about the task domain.

Answer to Review Question 1

Hierarchical task structures and plans.

Answer to Review Question 2

This statement is false. Cognitive Task Analysis techniques focus on the details of what is going on in the head and small-scale actions. GOMS is an example of a CTA technique. It has the same type of hierarchical structure as HTA, but has selection rules instead of plans.

Answer to Review Question 3

HTA, like any task analysis, start from data about how people perform tasks. This data may be gathered by observation, interview, or by reading documentation.

Answer to Review Question 4

Construction involves structuring and re-structuring the description (hierarchy and plans) to make it as meaningful and useful as possible. Stop expanding description when it is not useful to expand it further.

Answer to Review Question 5

There are many possible answers to this. The following is just one example.

- 0 process orders
  - 1 process one order
    - 1.1 receive order
      - 1.1.1 note pizza requirements

        - 1.1.1.1 note toppings required
        - 1.1.1.2 note price
        - 1.1.1.3 add to bill
      - 1.1.2 note additional requirements (e.g. drinks)
      - 1.1.3 note name and address
      - 1.1.4 take payment details
    - 1.2 make pizzas
      - 1.2.1 make a pizza to correspond to one order
        - 1.2.1.1 prepare dough
        - 1.2.1.2 add toppings as specified
      - 1.2.2 put pizzas in oven
      - 1.2.3 wait until pizzas cooked
      - 1.2.4 remove pizzas from oven
      - 1.2.5 put each pizza in a box
    - 1.3 deliver pizzas
      - 1.3.1 give pizzas corresponding to one order to delivery motorcyclist
      - 1.3.2 give bill and address to motorcyclist
      - 1.3.3 wait
      - 1.3.4 check payment received against bill and clear account
  - Plan 0: do 1 repeatedly (possibly in parallel)
  - Plan 1: do 1.1 – 1.2 – 1.3
  - Plan 1.1: do 1.1.1, 1.1.3, 1.1.3 and 1.1.4 in any order
  - Plan 1.1.1: do 1.1.1.1 – 1.1.1.2 – 1.1.1.3
  - Plan 1.2: do 1.2.1 repeatedly until order complete, then 1.2.2 – 1.2.3 – 1.2.4 – 1.2.5
  - Plan 1.2.1: do 1.2.1.1 – 1.2.1.2.
  - Plan 1.3: do 1.3.1 and 1.3.2 in either order, then 1.3.3 – 1.3.4

Answer to Review Question 6

KB analysis is concerned with hierarchies of objects and actions, and classifies items in terms of relationships such as class membership, attributes, and features (using AND, OR and XOR relationships).

Answer to Review Question 7

True. One way to structure a manual would be in terms of related actions or concepts, and KB analysis would help with this conceptual structuring.

Answer to Review Question 8

Entities include objects (concrete, composite and actors), attributes, actions and events. Note that there are many other terms (e.g. patient, agent, instrument) that are not entities in their own right, but are terms that define a relationship between entities. (If this is not immediately obvious to you, think hard about it).

Answer to Review Question 9

Task analysis focuses on domain entities, whereas ER analysis for database design focuses on the entities that are to be represented in the resulting computer system.

Answer to Review Question 10

Objects have attributes. Actors perform actions. Composite objects comprise (simpler) objects. Objects are located at objects. Actions have patients (objects), actors (objects) and instruments (objects). Actions may take place before events, be triggered by events, or caused by events.

Discussion on Activity 1

In designing your HTA, you will probably have included both searching and browsing as activities. The design consideration is likely to focus on browsing, and on how that can be made efficient and effective.

Discussion on Activity 2

Note that actions should not be described at the level of a button-click, but at the level of something that has some domain significance. Similarly, objects should not be described in terms of icons, but in terms of objects that have some domain significance. Depending on which sites you choose, you may or may not be able to use your informal analysis to propose some design improvements to the sites.

Discussion on Activity 3

As in exercise 4, you should be aiming to identify entities and relationships of various kinds. You should reflect on both what is involved in ER analysis, and what circumstances you would find it most and least useful in.