Chapter 7. Models of the User

Table of Contents

Context

Whenever something is built – be it a bridge, handheld Web browser or a new workflow application – models are likely to be involved. Models are used both to evaluate a design (have we made the right choices?) and in the construction process itself (i.e. helping us to generate designs and other artefacts). Models are important in interactive systems design and evaluation. In this unit you will look models and modelling techniques that focus on users. The unit begins by looking at user requirement modelling and continues by considering two example cognitive models (GOMs and KLM) that are used to capture and evaluate user information processing behaviour.

Review Question 1

Why are models important in the design process? Consider what advantages there are to using models.

In what ways do models contribute to the design process?

Answer at the end of the chapter.

Review Question 2

Both user centred requirements analysis and cognitive modelling employ models. Consider what the fundamental differences between the two approaches use models.

Answer at the end of the chapter.

Objectives

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

- Understand the key concepts concerned with modelling.
- Explain why and how models are used in interactive systems design.
- Explain why user-centred requirements modelling is more effective for interaction design than traditional methods.
- Describe several user-centred techniques that can be used to capture user requirements (e.g., Soft Systems Methodology & participatory design).
- Explain what is meant by a cognitive user model, give an overview of the range of models available and indicate the usefulness of such models
- Distinguish between the high-level (GOMs) and low-level (KLM) cognitive models.
- Produce GOMs and KLM analyses of simple user tasks.
- Show how GOMs and KLM can be used to analyse an existing system or proposed design.

Models of User Requirements and Context

There are typically several interested parties in the design and construction of systems. These include the clients who commission the development, the designers and developers who design and build the system, and their management who look after the running of the project. These people are primarily interested in what the system should do – its functional requirements. An important set of people who are often forgotten about are the users who will actually end up using the system. In order to build systems which users will find usable we need to consider their needs rather than exclusively concentrating on functional requirements of the system. These non-functional requirements include issues such as the usability of the system and how acceptable it is to the users. Without such considerations systems may be produced which have been designed without considering whether they would be acceptable to users e.g. the introduction of an automated ordering system at a restaurant may not be acceptable to the users (waiters) if it requires them to spend a large proportion of their time interacting with it rather than interacting with restaurant goers. Clearly, if users do not find a system acceptable they will be reluctant to use it and so extensive redesign or user retraining may be necessary costing large amounts of time and money.

The rest of this section looks at approaches which are considered user centred. That is, they focus their attention on the user and their non-functional requirements rather than concentrating solely on functional requirements.

Socio-technical Models

Socio-technical models consider the users within the organisational context that the system will be introduced into. The particular focus is on the interplay between the social and technical issues - hence the term socio-technical. One such approach is CUSTOM which concentrates on identifying who will be involved with the new system, and what their requirements are (not just functional requirements). Once these have been identified the designers of the system should have a better grasp of the non-functional user requirements as well as the organisational structure itself and how this relates to the design. Taking these requirements into account should help designers develop more acceptable and efficient systems.

CUSTOM refers to the people who would in some way be involved with the new system as stakeholders who can be classified in four ways:

- **Primary** – the people who will use the system e.g. the waiters in a restaurant.
- **Secondary** – people who produce input for the system, or receive output from the system, but do not directly use it e.g. the restaurant goers who are presented with a bill produced by the system at the end of their meal.
- **Tertiary** – people who are touched by the success or failure of the system, but are neither primary nor secondary stakeholders e.g. the owner of the restaurant chain.
- **Facilitating –** the people involved in the system's design, development, and maintenance.

Once the stakeholders have been identified their characteristics are analysed to develop user centred requirements for the system. Dix, in his 1998 book, summarises this process of stakeholder analysis in terms of the following questions:

- What does the stakeholder have to achieve, and how is success measured? E.g. waiters have to ensure diners are served at appropriate times and are happy with the level of service (not too intense or too disinterested). One way to measure a waiter's success may be the size of their tip.
- What are the stakeholder's sources of job satisfaction? What are the sources of dissatisfaction and stress? E.g. for a waiter this may be the pleasure of serving food and providing a pleasant eating atmosphere. They may be stressed by angry customers or a large number of customers to keep happy at the same time.
- What knowledge and skills does the stakeholder have? E.g. a chef has extensive knowledge of cooking which the waiters may not.
- What is the stakeholder's attitude towards work and computer technology? E.g. the owner of a chain of restaurants may be a technophile whilst a chef may be a technophobe. This may well cause conflict in the introduction of new technology.
- Are there any work-group attributes that will affect the acceptability of the product to the stakeholder? E.g. is there something about people who become waiters that will affect how well they accept the product?
- What are the characteristics of the stakeholder's task in terms of frequency, fragmentation, and choice of actions? E.g. a busy waiter will typically have to perform many fragmented tasks with high frequency in order to keep the diners happy.
- Does the stakeholder have to consider any particular issues relating to responsibility, security, or privacy? E.g. waiters may need to be discreet with regular diners who dine each night with a different partner, and may need to ensure that credit card payments are dealt with securely.
- What are the typical conditions in which the stakeholder is working? E.g. the chef of the restaurant typically works in a hot and dangerous environment whereas the owner of the chain of restaurants may work in a conventional office environment.

Note how little is asked about the technology and the functional requirements of the new system. The intention is to understand users' working context and their possible attitudes towards the new system so that it will hopefully be more acceptable.

Activity 1 - CUSTOM

A bank has asked a systems development company "Cash Machines R Us" to introduce new cash machines into one of their branches. Use the CUSTOM approach to socio-technical modeling to identify who the stakeholders are in the introduction of a new cash machine.

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

Soft-Systems Methodology

Soft-Systems Methodology (SSM) takes a broader view than socio-technical approaches by considering the organisation as a whole. From this perspective the stakeholders and technology are components of the larger context. As with other user centred approaches, the intention is to understand the context of the people involved with the system rather than making explicit design decisions. These models of the situation are instead used to inform design and help designers understand the context in which the final system will fit.

SSM involves the development of three kinds of models to help understanding the organisational context. The first kind of model is referred to as the rich picture which provides a detailed description of the problem situation. This includes details similar to the models of stakeholders developed in socio-technical approaches discussed previously – who the stakeholders are, what groups they work in, and what tasks they perform. In addition the organisational structure is modelled where it impacts on stakeholders' work. As this model provides the context for further models it should be clear and informative for designers. It could be developed from interviews with people in the organisation, observations of their work practices, or more interactive approaches such as workshops in which stakeholders act out situations in order to help explain their work context.

The next stage moves the focus of analysis from the real-world situation to the development of definitions of what stakeholders perceive to be the activities taking place in the organisation. These definitions are referred to as root definitions of the system. There may be several different root definitions – representing different stakeholders' perspectives – which need to be reconciled at a later stage. Root definitions model the following aspects:

- **Clients** – people who benefit or accept output from the system e.g. in our restaurant example the clients may be the diners who benefit from the restaurant nutritionally and receive output from the system by way of a bill.
- **Actors** – stakeholders who perform activities in the system e.g. the waiters and chefs in the restaurant.
- **Transformations** – what changes the system performs on things in the environment e.g. a system which produces bills in a restaurant transforms diners' requests for food (conveyed by the waiters) into bills by the end of the meal.
- **World view** – how the system is perceived by a client e.g. a waiter may perceive an automated billing system as beneficial as it reduces the work they have to do in maintaining the bills for multiple diners.
- **Owner** – who the system belongs to, and who can allow changes in the system e.g. the owner of the restaurant chain owns the automated billing system.
- **Environment** – what factors influence the system e.g. a restaurant has to abide by certain health and safety standards.

Finally a conceptual model is constructed which details what the system has to do to meet the root definitions. The most important part of the root definitions are the transformations which are used in the conceptual model to define what is achieved, and how it is achieved. These achievements are usually modelled hierarchically to provide different levels of detail – much as tasks are decomposed in task analysis (see Unit 8). For example, the overall achievement of serving a diner includes successfully finding out what the diner wants, serving them, clearing the table, and ensuring that the food is paid for. Considering the hierarchic structure, achieving payment for the food involves several sub-achievements such as producing the bill, collecting the money, and possibly producing a receipt.

The conceptual model is used to identify differences between the real-world situation and the model of how the stakeholders perceive the system. These differences can then be used to inform change and/

or development of appropriate systems. The key outcome of the whole SSM approach is for designers to have a better understanding of the context in which developed systems are to be placed.

Activity 2 – Soft Systems Methodology

For the example in Activity 1 construct a root definition for the new cash machine from the perspective of a bank customer who wants to withdraw money.

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

Review Question 3

Discuss the similarities (and differences) between socio-technic and soft systems methodology approaches to understanding user requirements , what sorts of things do they look at, and how can this provide input into design?

Answer at the end of the chapter.

Participatory Design

The two approaches we have looked at so far develop user centred models which are typically used in the early stage of system development. In contrast, participatory design aims to keep the whole process of developing a system user centred. This is achieved by including users in the design team rather than treating them as subjects of analysis who remain outside the core design situation. The motivation for this is that users are essentially experts on their work situation – they know full well what work they do, how the organisation works for them, and who they need to work with – and so including them in the design process is essential if the new design is to fit well into their context. Moreover, the introduction of new systems typically changes the work context. For a new system to be acceptable these changes in the work context also need to be acceptable. Having users in the design team should help ensure that these changes do not overly disturb other users.

Participatory design has three main characteristics:

1. **Work focused** – design concentrates on improving the workers' environment and tasks they perform rather than focusing on the system requirements.

2. **Collaboration** – the designers and users collaborate on the design so that the users can contribute at every stage.

3. **Iterative** – design does not just happen once, the emphasis of participatory design is on several design and evaluation stages which build to a final design.

As participatory design focuses on the collaboration between users and designers it needs to employ various techniques and models to communicate ideas between them including:

- **Brainstorming** – sessions in which users and designers generate a range of ideas which are developed without judgement. These ideas are then fed into other techniques to be developed further or dropped.

- **Storyboarding** – a rough idea of a user's activities can be presented via a storyboard. The storyboard models user activities as a series of drawings – a little like a comic strip description of the activity. They are simple to develop and help users communicate with the designers about what they do, and how they do it.

- **Workshops** – these provide a forum for discussion in which designers and users can ask each other about their perspectives and so establish common understandings of the design issues as well as focusing their views of the design. Typically they are used to fill in gaps in understanding about the situation. As such the designers usually enquire about users' work environment, and the users ask about technological possibilities.

- **Pencil and paper activities** – this provides a more interactive way of talking through designs than storyboards. Typically, rough designs are drawn (i.e. possible designs are modeled using pencil and paper) and then users consider how they would use it by moving through the design step by step. Problems with the design can then be identified from trouble the users have as they move through it. These problems can then be addressed in the next iteration of the design.

Participatory design was developed in Scandinavia where it has been widely accepted. However, the time and effort it requires, and reliance of less hierarchically structured organisations (where designers and users are treated equally), has meant that its use is less wide spread outside Scandinavia.

Activity 3 – Participatory Design

For the example in Activity 1 construct a storyboard of a customer withdrawing money from a cash machine. Imagine that the cash machine has several pages of information which it displays. For any operation the first page allows the user to enter their PIN (Personal Identification Number). If this is correct for the customer's card the machine shows the next page which allows the user to select one of several services on offer. If they select the withdraw cash service they are presented with a page from which they can select a predetermined amount of cash, or can select an option to allow them to determine how much they want. If they select this option they are then presented with a page which allows them to enter a value up to £250. Once the amount of money has been entered (either by selecting a predetermined amount or entering their own amount) the machine returns the card and then the cash to the customer.

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

Review Question 4

How does participatory design relate to approaches such as socio-technical and Soft Systems Methodology. Concentrate on what they aim to do, and what sort of models they use.

Answer at the end of the chapter.

Summary of User Requirements Modeling

This section looked at models of user requirements. These are representations of the users' world which are used to inform the understanding of non-functional requirements such as whether the developed system would be acceptable in the workplace. We looked at socio-technical models which consider who the stakeholders in the development of a system are. We then considered soft systems methodology which takes a broader view as it also models the organisation in which the stakeholders work. Finally we touched on participatory design which includes users in the design process and uses models to help designers and users understand each others' points of view. All of these approaches involve extra time and effort than simply defining the functional requirements of a system. However, systems need to be developed which take into account users' needs and work context otherwise they may be unusable or unacceptable to the intended users. It is the effectiveness of user centred approaches to take this into account that makes them attractive to developers.

Cognitive Modeling

As we discussed at the start of this unit, a model is a representation of something that we can easily work with. In the previous section we looked at models of user requirements – essentially models of users' context. In contrast, this section considers models of users' cognitive abilities. That is, rather than looking at the context in which users work we will consider models of how people behave. These cognitive models are important because they provide a way of understanding people's cognitive abilities which can be used to inform design and, as we shall see in Unit 9, to evaluate systems. In this section we look at two cognitive models: GOMS and ICS.

GOMS

GOMS uses an explicit model of cognitive processes developed from studies of users. GOMS itself stands for Goals, Operators, Methods, and Selection rules. This core set of concepts describes the tasks that users perform, and how they achieve them. GOMS is more than just a way to describe tasks though, it provides a family of modelling techniques which can be used to generate predictions of the time to complete tasks given descriptions of the tasks and the user interfaces to be used. These predictions are based on a simple model of cognition called the Model Human Processor (MHP) and have an absolute accuracy of between 10% and 20%. The basic assumption of GOMS is evident in its name – that users understand their goals, GOMS considers the actions to meet such goals. Having made this assumption GOMS can give us feedback on the coverage and consistency of the user interface. By coverage we mean whether the user interface contains the necessary functionality to support the tasks it was designed for. Consistency refers to whether similar tasks are performed similarly with the user interface. Moreover, GOMS can give an indication of whether frequent goals can be achieved quickly using the given interface.

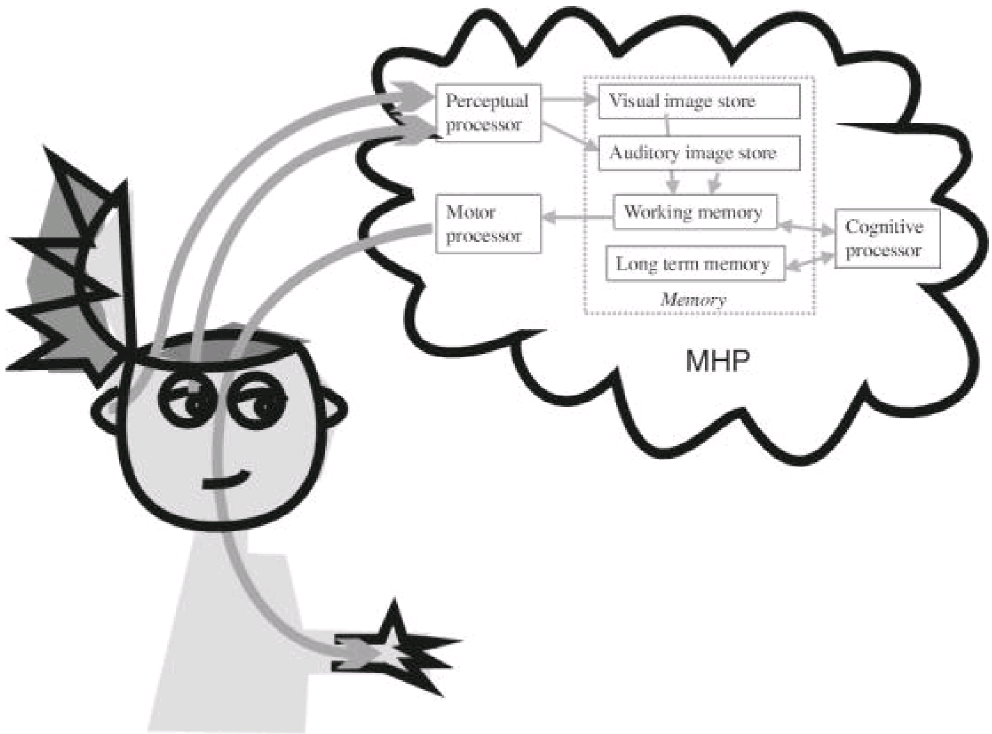
Model Human Processor

The central part of GOMS (Card, Moran, and Newell, 1983; see Chapter 6 of Dix 1998 for a summary) is the Model Human Processor (MHP). This is a simplified view of the psychological processes involved in human cognition which can be used to make approximate predictions of user behaviour. Card et al. based this model on empirical evidence i.e. psychological studies of users. Even though it is based on psychological evidence and theory the motivation for the MHP is that is should be usable by non-psychologists e.g. designers and evaluators. The MHP is consists of:

- **set of processors –** systems which process information and may make decisions
- **set of memories** – areas in which information is stored in various forms by processors
- **set of principles of operation –** these guide the operation of the processors and are crucial to a realistic account of human behaviour

The MHP itself is divided into 3 interacting subsystems: the perceptual system, the motor system, and the cognitive system. Each of these subsystems has their own memories and processes and works on different kinds of information as illustrated in the following diagram.

1. **Perceptual system** - input from the eyes and ears are stored symbolically in visual and auditory image stores which go on to be processed in working memory by the cognitive system, and possibly stored in long term memory for later access. E.g. the sound of a car horn is coded as a loud warning.
2. **Cognitive system** - processes information from working and long term memory to make decisions about how to respond. E.g. processing the fact that there is a loud warning noise, recalling that evasive action should be taken in such situations, and deciding to run to the pavement.
3. **Motor system** - executes responses to decisions made by the cognitive system. E.g. running to the pavement.



Each component of these systems has parameters which are used in the generation of behavioural predictions. Each memory system is defined in terms of its storage capacity (how many pieces of information can it hold at one time), and the decay time of an item (how long is it before an item is forgotten). Similarly, each processor is defined in terms of processor cycle times – how long it takes to process one piece of information. Card et al. (1985; see Johnson, 1992, for a summary) defined approximate values for these parameters based on psychological research which are listed below. Note that these values assume skilled (errorless) performance and only predict time for responses to stimuli – not for self initiated actions.

Table 7.1. Memory parameters

Memory type	Storage capacity	Decay time
Auditory	4 letters	1500 msec
Visual	17 letters	200 msec
Working memory	7 chunks of information	7 sec
Long term memory	Unlimited	Never

Table 7.2. Processor parameters

Processor type	Processing cycle time
Processor type	100 msec
(eye movement)	230 msec
Cognitive	70 msec
Motor	70 msec

So, we have described the MHP, but we need to return to the core of GOMS – the goals, operators etc. – before we can see how these parameters are used to make predictions.

Goals

Goals are something that the user wants to achieve e.g. go to airport, delete a file, or create a directory. They have a hierarchical structure – that is they are composed of many sub-goals which need to be achieved to meet the larger goal. These are similar to the goals identified in Task Analyses (see Unit 8).

Operators

Operators are elementary (can not be decomposed into smaller operations) perceptual, motor or cognitive acts which are necessary to change user's mental state or environment. As such they are the lowest level of a GOMS analysis. Using GOMS a user's behaviour can be recorded as a sequence of operators as operators can't occur concurrently. They are a similar level of description as actions in task analysis (see Unit 8).

For example, to move a file to a different folder the user might perform the following operations:

- Move cursor to item
- Hold mouse button down
- Locate destination icon
- Let go of mouse button

Methods

From operators we build up methods which are sequences of steps that accomplish a goal (and so are like tasks in a task analysis (see Unit 8)). As with goals these methods can include other (sub) goals. A fundamental assumption in GOMS is that methods are learned and routine (so no problem solving involved), and that there is only one way a user stores knowledge of a task.

For example, a user moving a file to a different folder could be described in GOMS as:

- **Goal** – move file to a different folder
- **Method** – move file
- **Operators** - Move cursor to item, Hold mouse button down, Locate destination icon, Let go of mouse button

Selection Rules

If there is more than one method to accomplish a goal, the Selection rules tell you which method to use. Again, as with methods, they assume error-free performance (so the user does not selected the wrong method by accident). They are written as IF … THEN statements as below:

```
IF <condition> THEN accomplish <GOAL>

For example:

IF <restaurant accepts credit cards> THEN <pay by credit card>

ELSE

IF <restaurant accepts cheques> THEN <pay by cheque>

ELSE

<pay by cash>
```

Activity 4 – GOMS

For the example in Activity 3 construct a GOMS model of a customer withdrawing money from a cash machine.

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

Keystroke Level Model

The lowest level of GOMS analysis is called the Keystroke Level Model (KLM). This produces quantitative predictions of the time it would take a skilled operator to complete a task. Again, it assumes error-free performance by the operator.

Execution of a task is described in terms of

- 5 physical-motor operators:
  1. **Tk**: (k)eying – how long it takes to press a key (including using modifiers such as the shift key)
  2. **Tp**: (p)ointing – how long it takes to move the mouse (or other such input device) to a target
  3. **Th**: (h)oming – how long it takes to change between input devices e.g. changing between mouse and keyboard
  4. **Td**: (d)rawing – how long it takes to draw a line using an input such as a mouse
  5. **Tb**: click (b)utton – how long it takes to click the mouse button
- **Tm**: (m)ental operator – how long it takes to perform the mental processing for the task
- **Tr**: system (r)esponse operator – how long the system takes to respond

Therefore, execution time for a task is described in terms of the sum of the operators used. For example, suppose we had typed the sentence the quick fox jumps over the lazy dog. Now we want to insert brown just after quick, using a word processor, and assuming that the current point is at the end of the sentence, we need to perform the following steps:

1. move hand to mouse
2. position mouse just after quick
3. move hand to keyboard
4. formulate word to insert - brown
5. type brown
6. reposition insertion point at end of sentence

In terms of the KLM the following operators are needed for the above steps:

1. H (mouse)
2. P, B
3. H (keyboard)
4. M
5. K (b) K (r) K (o) K (w) K (n)
6. H (mouse), M, P, B

So, in total the execution time for this simple task is 3Th + 2Tp + 2Tb + 2Tm + 5Tk (assuming there is no significant response time for the system). Card et al. derived values for the time to complete these operators from empirical studies. These are listed below (for an expert typist), and give a total execution time of 1.2 + 2.2 + 0.4 + 2.7 + 0.6 = 7.1s in this case. As we shall see later in Unit 9, these quantitative predictions of execution time can also be used to compare designs.

Operators	Time (s)
Tk	0.12
Tp	1.10
Th	0.40
Td	1.06
Tb	0.20
Tm	1.35

Activity 5 – KLM

Carrying on with the example in Activity 4, imagine that customers could withdraw money using their personal computer. In this case data entry would be via the keyboard, and selection of options would be done using the mouse. Using KLM, work out the execution time for the activity of withdrawing £10 assuming that both keyboard and mouse are to be used, that the PIN is 1234, that it takes the system 10s to return the card and cash, and that £10 is one of the predetermined amounts listed.

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

Summary of GOMS

GOMS provides a way of making predictions about the time an expert user would take to complete a task using a given user interface. Furthermore, as GOMS modeling makes user tasks and goals explicit these descriptions could be usefully re-employed in the development of an on-line help system. These descriptions can additionally be used to suggest questions users will ask and the answers in terms of actions needed to complete tasks and meet goals.

However, as mentioned several times before, the tasks must be must be goal-directed, that is the user must have a specific aim in mind. Some activities are more goal-directed than others, but it could be argued that even creative activities contain goal-directed tasks. Furthermore, GOMS assumes that tasks involve routine cognitive skill as opposed to problem solving, and that no errors occur, which is hardly realistic.

Review Question 5

GOMS is a cognitive modelling approach. What does it model, and how?

Answer at the end of the chapter.

Interacting Cognitive Subsystems

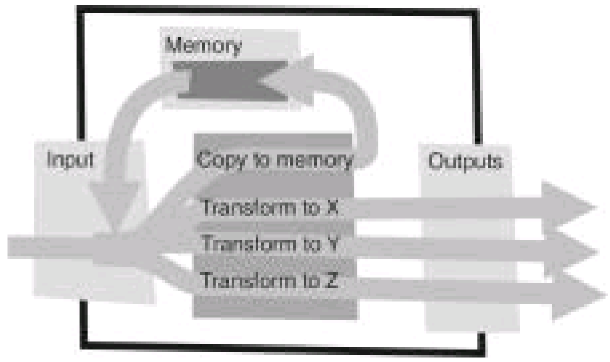
ICS is an elaborate framework which assumes that human perception, cognition, and action can be analysed in terms of discrete, inter-linked, information processing modules. In contrast to GOMS, ICS is a much richer way of modelling human cognition as we shall see in this section.

Subsystems

Each subsystem of ICS is independent and operates in a specific domain of processing of which there are three main components:

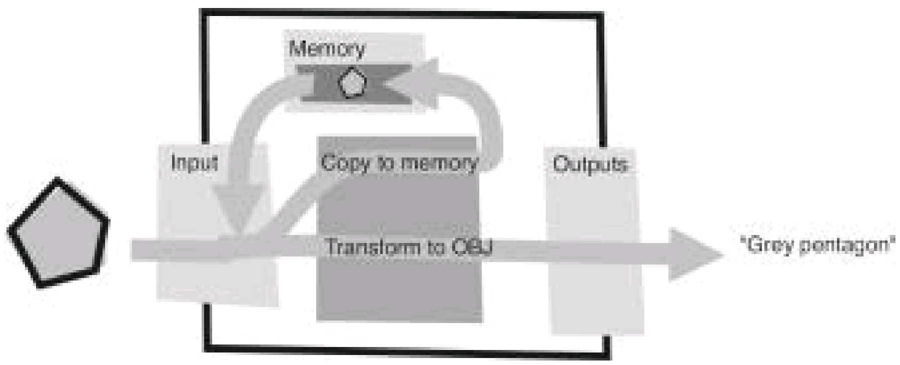
- **sensory** – visual and auditory stimulus
- **representational** – representations of information
- **effector** – body movement

Although each subsystem operates in different domains, and on different kinds of information, they all share a common structure as illustrated in the following diagram. Each subsystem has an input to its left, and one or more outputs to its right, as well as its own memory record and a set of transformations. All information which is input to the subsystem is stored in the memory record (and can at a later point be used as an input to the subsystem). The outputs depend on the transformations which transform information from one form to another. In the example we have three transformation which transform the input information into three different codes – X, Y, and Z.



An ICS Subsystem

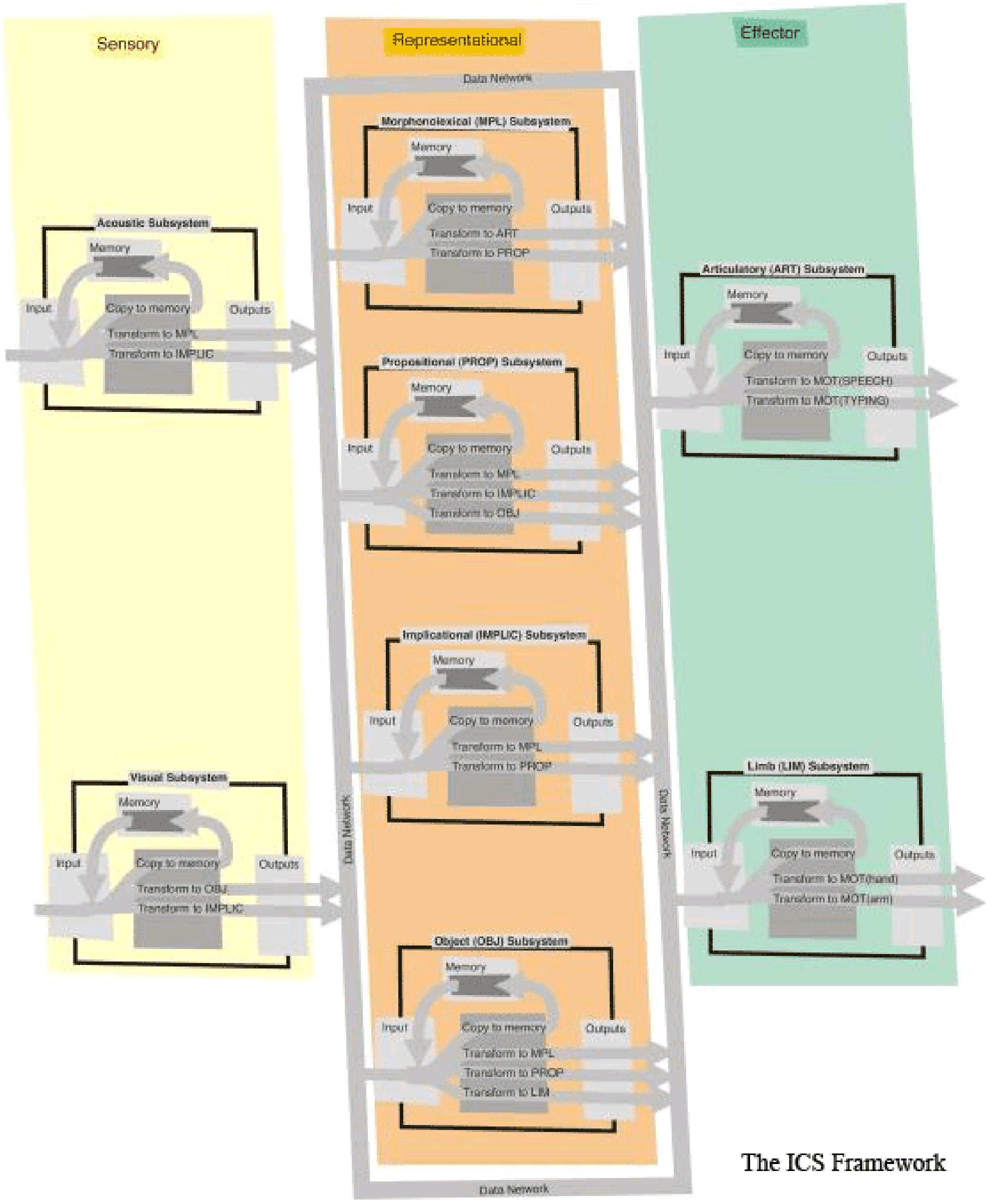
As mentioned previously, ICS models human cognition in terms of several of these subsystems linked together. We shall come to the complete network later, but let's start to understand the ICS framework by considering the operation of one subsystem. A good point to start understanding ICS is by considering the visual subsystem (illustrated below) which forms part of the sensory domain. This takes input from the eyes in the form of sensory representations with information about edges, angles, shades, contrasts, hues etc. (here the person is looking at the pentagon to the left of the diagram) and transforms it to object representations with information about depth, position, orientation, shapes, etc. (here the object representation describes the image as a grey pentagon). Note that a copy of the visual representation is stored in the memory record for later use. Note also that the clarity of the visual representation alters the success of the transformation to an object representation. It is important to remember that these two representations (visual and object) are both mental representations, but at different levels of information.



The Visual Subsystem

The key to using ICS is the inter-linking of the subsystems. Below is a diagram of the complete ICS framework – the VIsual Subsystem (VIS) just described can be seen in the bottom left of the diagram in the sensory domain i.e. input from the sense. We need to further process information to extract meaning and make decisions e.g. we may need to interpret the grey pentagon as a fifty pence piece and then use it appropriately.

ICS can be used to describe a user's mental processes whilst performing a task. Such descriptions can then be used to assess the relative amounts of cognitive resources (the numbers of transformations) needed to complete tasks with different interfaces. In the example a user is revising some text in a word processor. Whilst reading the text they notice a particularly difficult sentence and decide to rephrase it. In an ICS description of the cognitive processes involved in achieving this task we might start by considering what the sensory inputs are. As the user has read the text from the screen they are using their visual sensory input. So the first subsystem involved is the Visual Subsystem which transforms the visual representation into an object representation (VIS ® OBJ). This transformation produces information about what letters are being viewed. As the output is in OBJ form it is then processed by the Object Subsystem which performs the transformation OBJ ® MPL. This morphonolexical representation encodes the surface structure of the sentences in a speech based code which discards the surface structure of the words i.e. it encodes what the words are, rather than what letters they are composed of. The MPL representation is then processed by the Morphonolexical Subsystem which performs the transformation MPL ® PROP producing a representation which encodes the meanings of the individual words in the sentence, and the relationships between the words. From this representation the Propositional Subsystem transforms it to an implication representation (PROP ® IMPLIC) in which the meaning of the sentence itself is identified. Several different meanings of the sentence may be identified in which case there may be several iterations of PROP ® IMPLIC (Propositional Subsystem) and IMPLIC ® PROP (Implicational Subsystem) to arrive at a satisfactory understanding of the meaning. At this point the representation needs to be transformed into a form suitable for the Limb Subsystem so that the correction can be made. This would involve the following transformations by Subsystems: IMPLIC® PROP (Implication Subsystem), PROP ® OBJ (Propositional Subsystem), OBJ ® LIM (Object Subsystem), LIM ® MOT(hand) (Limb Subsystem – transforms LIM into actual hand actions).



The ICS Framework

Of course, there is plenty more processing involved in locating the text on the screen, positioning the cursor, reading the menu bar etc. but this description gives and idea of the processes involved. The sequence described can be summarised as follows (Þ indicates external input or output, ® is a transformation, » indicates data transmitted across the data network, and {} enclose a set of transformations which may occur several times):

```
Þ VIS ® OBJ »

» OBJ ® MPL »

» MPL ® PROP »

{» PROP ® IMPLIC »

» IMPLIC ® PROP »}

» PROP ® OBJ »

» OBJ ® LIM »
```

```
» LIM ® MOT(hand) Þ
```

Summary of ICS

ICS is intended to provide a detailed description of the cognitive resources needed to use and learn a system. As such it is far more powerful than GOMS, but also much harder to use and to interpret the results it produces. Even trained psychologists would find it hard to consistently determine which Subsystems are involved in users attempting to complete tasks. One approach to alleviating this problem is to develop some sort of software toolset to help evaluators use the framework. This might support activities such as determining which Subsystems are used, and which representations are appropriate.

Review Question 6

ICS provides a detailed model of human cognition which can be used to determine how much cognitive effort a user will have to employ to complete a task. How does this differ to GOMS in terms of its model of human cognition, the intended users of the model, and the kinds of assessments it can make?

Answer at the end of the chapter.

Activity 6 - ICS

ICS is notoriously difficult to use. How would you develop a tool to make its use easier , what functionality would it have, and how would that make using ICS easier?

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

Summary of Cognitive Modeling

In this section we looked at models of human cognition which are intended to help designers understand how users would behave with their systems. Both models have a computational feel reflecting their roots in cognitive psychology. Furthermore, both models can be quite difficult to use, especially for large systems

Summary of User Modeling

The two approaches discussed in this unit – user requirements modelling and cognitive modelling – are very different uses of models. The first models the situation in which users work and the constraints their environment places upon them, and therefore new systems, whereas the second tries to give designers some understanding of the working of the human mind. They may be radically different models, but the reason for their existence is the same – to ensure that designs take account of users. This is referred to as user centred design and is an important departure from the conventional approach to design which concentrates on functional requirements of the system. The aim is that by considering the user (either from the point of view of their work context, or their cognitive abilities) the developed systems will be successfully deployed in the workplace and accepted by the users.

Discussion Topics

There are many new concepts in this unit. If you want to discuss with your colleagues or make comments about the concepts to them, use the on-line facilities.

Discussion topics:

- User centred requirements analysis increases the effort of developing systems – what benefits can such approaches bring.

- How would you convince the company you work for to adopt user centred approaches?
- How can some of the more complex approaches (e.g. GOMS and ICS) be made more accessible to evaluators?

Answers and Discussions

Answer to Review Question 1

Models provide us with a representation which is easier to use and manipulate than the thing we are designing (that the model represents). This means that it is easy to test out different designs to see which is best for a given purpose. We can also use models to model a situation e.g. the social situation in which the thing we are designing will be placed. This helps us to understand how the new system will impact on the current working situation. Finally, we can use models to provide approximations of human behaviour which can be used to determine whether our designs will be difficult to use, without involving users.

Models contribute in two main ways – generative or evaluative. Evaluative allow us to assess whether the final product will meet some criteria. Generative models provide input to design – it should be noted that most models have some generative aspect to them.

Answer to Review Question 2

Use centred requirements analysis employs models to represent a typical user’s work environment. This ranges from representations of the stakeholders and their perspectives to considerations of how systems influence the social and organisation situation. Cognitive models, on the other hand, provide a simplistic view of how user might react to situation, and in turn, how they might use a newly designed system.

Answer to Review Question 3

Both socio-technic and SSM model the stakeholders involved in the development of a new system. These models give designers a grasp on the context of who are involved in the development of the new system, and what impact this will have on their working environment. The primary difference between socio-technic and SSM approaches is that SSM takes a wider view of the context. This means that it provides a model of not just the stakeholders, but also the environment in which they work. Providing such a wider view allows designers to get a grasp on the wider picture, but does involve extra effort on the designers’ and analysts’ part to develop and understand the models.

Answer to Review Question 4

The crucial difference between participatory design and socio-technical and SSM approaches is that participatory design is a design philosophy in which users are considered central and are included on the design team throughout the stages of development. Socio-technical and SSM approaches are concerned with establishing models of user requirements (as opposed to necessarily including them in the design process). In contrast, models used in participatory design are used to support communication between designers and users e.g. storyboards which model typical user interactions.

Answer to Review Question 5

GOMS uses a Model Human Processor to model human cognitive processes. This is used in conjunction with models describing Goals, Operators, Methods, and Selection rules to determine execution times using a Keystroke Level Model. This provides predictions of task execution time for tasks that involve no user errors or problem solving.

Answer to Review Question 6

GOMS' model of human cognition is referred to as the Model Human Processor. Both models assume that there are interrelated information processors, but ICS' model is much more complex. Essentially the representational domain of ICS (containing 4 subsystems) is the cognitive processor of GOMS.

GOMS attempts to make quantitative predictions of task execution time by expert users. This takes no account of errors, learning, or problem solving. ICS, on the other hand, can model these issues, but does not give simple quantitative predictions. As such ICS is intended to be used by expert psychologist evaluators, whereas GOMS was intended to be used by trained evaluators.

Discussion on Activity 1

The following is a definition of the stakeholders concerned with the introduction of a new cash machine:

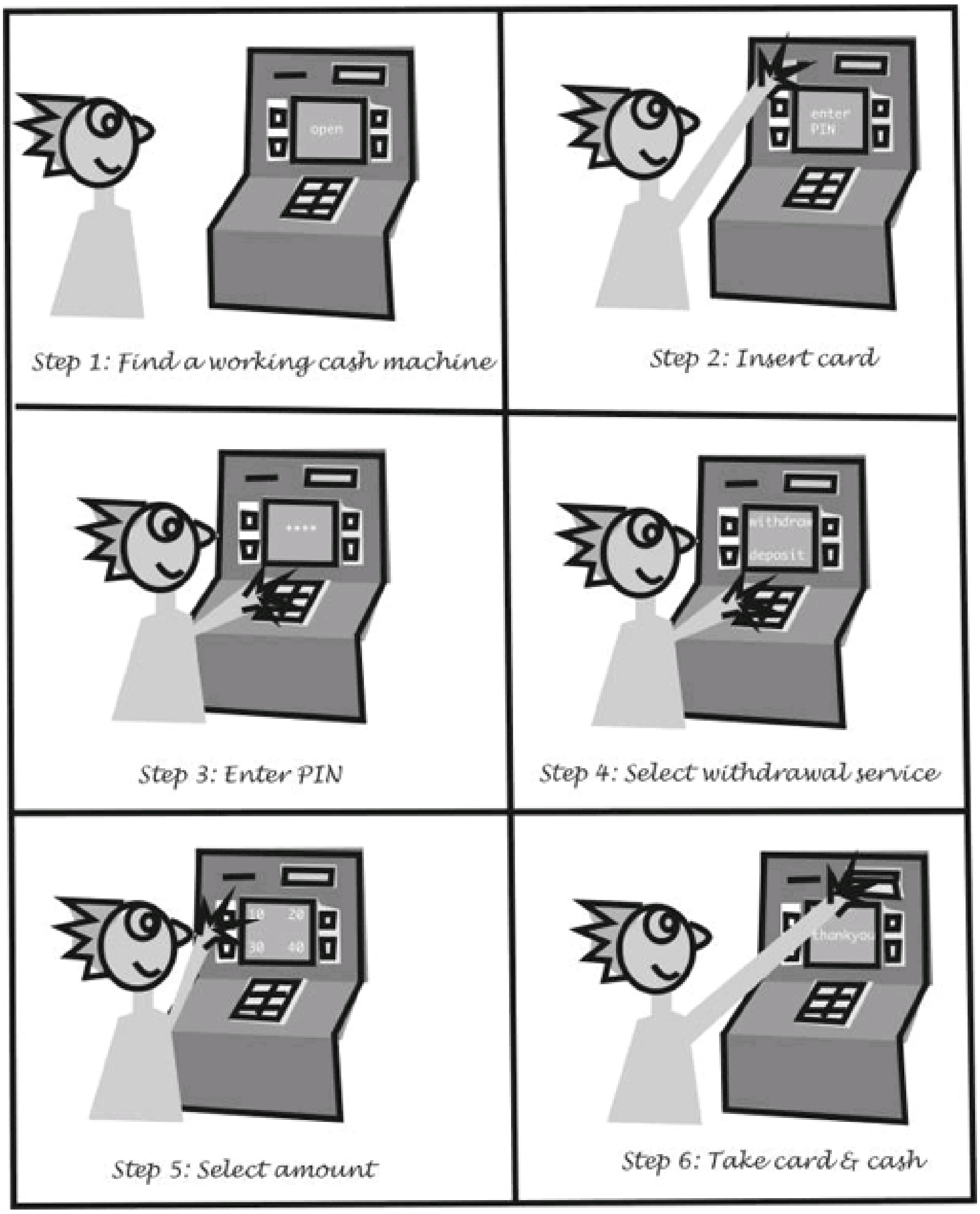
- **Primary** – people with cash cards, employees of the bank who have to service the machine e.g. fill it with money.
- **Secondary** – people such as accountants who receive receipts from the machine, people who receive money from the machine in payment e.g. shop keepers.
- **Tertiary** – managers of the bank and the directors of the bank.
- **Facilitating** – the employees of 'Cash Machines R Us' who are involved with the design, development, and later maintenance of the cash machines.

Discussion on Activity 2

- **Clients –** bank customer, bank clerks, bank manager, bank's board of directors.
- **Actors** – bank customer, bank clerks.
- **Transformations** – for withdrawal of money – transforms a customer's request for cash into cash and possibly a receipt.
- **World view –** for a bank customer **–** makes it quick and easy to withdraw money at any time of the day.
- **Owner** – owned by the bank, change may be authorised by senior people in the bank such as the board of directors, or maybe the head of information technology for the bank.
- **Environment** – the cash machine must be secure, robust enough to withstand all weather conditions and vandalism, and at a height that is convenient for most customers.

Discussion on Activity 3

A storyboard for this activity is as below:



Discussion on Activity 4

```
Goal – withdraw cash.

  Method - Enter PIN

    Operators – press-1st digit of PIN, press-2nd digit of PIN, press-3rd digit

  Method - Select withdrawal

    Operator – select ‘withdraw cash’

  Method - Select amount

    IF <amount is shown> THEN <Operator - select amount>

    ELSE
```

```
      <Operators - select ‘other amount’, press-1st digit of amount, press-2nd
    IF <3 digit amount of cash> THEN <Operator – press-3rd digit of amount>
  Method - Collect card & money
    Operators – take cash, take card
```

Discussion on Activity 5

GOMS	**Operator**	**Time(s)**
Goal – withdraw R100.		
Method - Enter PIN		
Operators – press-1, press-2, press-3, press-4	4xTk	0.48
Method - Select withdrawal		
Operator – select ‘withdraw cash’	Th + Tp + Tb	1.70
Method - Select amount		
IF <amount is shown> THEN <Operator - select amount>	Tp + Tb	1.30
ELSE		
<Operators - select ‘other amount’, press-1st digit of amount, press-2nd digit of amount>		
IF <3 digit amount of cash> THEN <Operator – press-3rd digit of amount>		
Method - Collect card & money		
Operators – take cash, take card	Th + Tr	10.40
Total(s)		13.88

Discussion on Activity 6

One approach would be to explicitly model each subsystem of ICS and connect them via data pathways. Analysts could then feed in different kinds of information to the sensors, indicated which subsystems they think might be used, and then observe how the data flows around the network. This would reduce the burden of working out how the data flows around the system.

A more comprehensive approach would be to use an expert system (a system which contains a lot of knowledge about an area, and some artificial intelligence) to help work out which subsystems might be involved in processing information. This helps reduce the problems of deciding which subsystems and representations are likely to be used.