0 An Investigation of the Learning of a Computer System.

Learning a Computer System

Michael Wilson, Philip Barnard,Allan MacLean
MRC Applied Psychology Unit
Cambridge U.K.

To appear in:
P. Falzon, N. Streitz, Y. Waern and J-M. Hoc (eds.) **Cognitive Ergonomics,** Cognitive Science Series. Springer: New York.

1. Introduction

It is widely acknowledged that the behaviour of users and their underlying cognition is complex. For various purposes, it may be necessary to provide a detailed characterisation of user behaviour and cognition which is unlikely to be derivable from a single theory or assessment technique. Consequently, a number of specific assessment techniques and theoretical interpretations must be matched to specific purposes. In order to match these two, it is necessary to understand the properties of the possible theoretical interpretations and assessment techniques.

Assessment techniques can be used either for research purposes, to provide information about how users represent and process information, or to assess commercial products as part of the design process. In the design process, this assessment can be at an early stage, in order to provide ideas, or late, in order to locate user difficulties. If one wishes to obtain quick results with minimal analysis effort in the design process there must be a conscious trade-off between the effort to complete any test and the speed and amount of information provided. Therefore, it may not be feasible to apply the full range of detailed and laborious measures used for research purposes. Nevertheless, the interpretation of research tests and the relationships between them can provide criteria for selecting which tests may be appropriate, and their validity for the more practical problems of designing and assessing a system.

This paper considers different assessment techniques in the context of a study of users learning a commercial integrated office system. The purpose of the study was to trace the development of different aspects of the cognitive processes and representations which support user performance during the learning of a system. Since it is acknowledged that no single assessment technique could deliver this, several different techniques were used. In order to incorporate the results delivered by these different techniques, it is necessary to develop explicit models of what each test delivers, and the view of the user onto which such results will be mapped. An attempt will be made in this paper to be explicit about both.

The initial view taken of user learning is characterisable by a general model of skill acquisition (after Fitts, 1964). This model divides learning into three phases. In the first, users acquire sufficient fragments of knowledge about a system to support the performance of some tasks. In the second phase, the knowledge recruited to perform tasks is ’compiled’ into procedures. In the third phase, users draw on these compiled procedures and exhibit performance which can be considered ’expert like’. A further distinction exists in this model between the accessibility of the representations of knowledge in the first and third phases. It is assumed (after Anderson, 1983) that the knowledge in the first phase is accessible to the processes of verbalisation, whereas the compiled procedures drawn on in the third phase would not be articulatable. When task performance is described by users who have reached the third phase of learning for those tasks, the explanation will not be based on the ’compiled’ knowledge which actually supports performance, but on other knowledge that they hold about the system. This knowledge may be consistent or inconsistent with the knowledge compiled into procedures, but may be drawn on during performance of novel tasks which are not proceduralised.

The assessment techniques used in this study embody three basic strands: performance measures; measures which allow access to articulatable knowledge; and measures which implicitly require subjects to use knowledge about a system. There were overlaps and differences in the aspects of performance and cognition captured by the tests used in the study. This paper will outline the techniques used, and illustrate both the similarities and distinctions in the results, in order to understand what aspects of cognition and behaviour each test is best at capturing. In particular, the view of learning which can be captured by these tests will be described.

2. Details of the Study

In the study a sample of 8 naive computer users learned how to use the VisiOn[1] interactive office system. This system incorporates three principle environments for word processing, graph drawing and spreadsheet calculation. These environments have a common conceptual interface in which users invoke command operations by pointing with a mouse at command words in linear menus. Once a command is invoked users follow prompts for the required mouse and keyboard actions (see Lemmons, 1983, for further details of the VisiOn system).

The study consisted of eight sessions which were systematically structured to enable the training, exercising and testing of, a broad sample of VisiOn functionality. The overall design of the study is presented schematically in Figure 1.

In session one, users familiarised themselves with the keyboard and mouse as well as the basic concepts associated with windows and menus by using the in-built tutorial within the VisiOn system. Users then began a systematic training programme. The training was by example and illustration. Users were shown a videotaped example of the method for each task, asked to perform the task they had seen using the same materials as in the videotape during which time they could ask the experimenter about any details of which they were unclear. After a block of three tasks had been watched and copied, subjects practised each task three times on new materials. Materials were presented to subjects as manuscripts with editing changes marked on them. As the sessions progressed more and more functions were illustrated, copied, and practiced.

There were 4 different performance measures (called the ’A’, ’B’ and ’C’ set tests, and a data entry test). Although these included subjects verbalising occasionally, there was one explicit test of verbalisable knowledge: the Prompted Knowledge Elicitation test (PKE). There were two tests that implicitly required the use of the subjects knowledge of the system, a questionnaire study and a task which involved the sorting of command terms into the system hierarchy. Details of individual tests are given later.

FIGURE 1. The structure of the testing sessions in the observational study on VisiOn.

SESSION					
1	TUTORIAL	VIDEO PRACT A1			
2	VIDEO PRACT A2	A SET			
3	VIDEO PRACT B1	A SET			PKE
4	VIDEO PRACT B2	A SET	B SET 1		
5	VIDEO PRACT B3	A SET	B SET 2		
6		A SET	B SET 3	C SET	PKE
7	RECALL, QUESTIONNAIRE, DATA ENTRY TEST				
8	DATA ENTRY TEST				

Key: Video Pract - watching a video of command sequences, imitating them, and then practicing each command sequence. A1 & A2 for A set test. B1 for B set 1, etc ...
A, B & C SETS - three performance tests.
PKE - Prompted Knowledge Elicitation test.
RECALL - A command name sorting test.

The training was arranged with sessions 1 and 2 on consecutive days and the remainder at intervals of 3 to 7 days. The duration of each session was fixed at 2 hours since this was thought to be a reasonable sample period for the intended users of such systems.

[1] VisiOn is a trademark of VisiCorp

3. Performance tests

Each of the performance tests in this study was selected to answer specific questions about the process of learning. Firstly, how does performance improve with practice and is there any evidence of the proceduralisation expected as a result of the general model of expertise? Secondly, do users incorporate new functions more easily as a result of the accrual of general system knowledge? Thirdly, how does this knowledge generalise to untaught functions? Fourthly, when users reach a state of expertise in a specific task, do they make judgments as to the most appropriate method to perform that task on the basis of the time to use the possible methods, or on the basis of other knowledge?

The general view of learning outlined above suggests that the information that supports behaviour passes from being declaratively represented to being procedurally represented. During this change in the representation of knowledge, there is an hypothesised parallel reduction in the time to perform the task and the number of errors during performance.

This change in performance time has been formalised in the ’power law of practice’ which states that the time to perform a task decreases as a power-law function of the number of times the task has been performed. It has recently been argued that this law not only applies to the domain of motor skills where it was originally recognised (Snoddy, 1926), but to a full range of human tasks (Newell and Rosenbloom, 1981) including perceptual tasks such as target detection (Neisser, Novick and Lazar, 1963) and purely cognitive tasks such as supplying justifications for geometric proofs (Neves and Anderson, 1981). Consequently, it is often assumed that as experience with a computer system increases, the general task performance time and error count will reduce.

The ’A set’ test was designed to assess the general reduction in performance time and errors for frequently used commands. A detailed report of this test is given elsewhere (Wilson et al, 1985a, Wilson et al 1985b), so only a brief summary will be presented here. In the A set test, subjects first imitated and then practiced nine basic command sequences during the first two sessions, and then used these commands in later sessions. Three command sequences were selected from each of the three environments in the VisiOn system. The commands were: for word processing: delete a word, move a block of text, enter an item of text; for spreadsheet use: enter a number, enter a label, enter a subtraction or summation formula; for graph plotting: select data for plotting, select a style for plotting (a line graph), plot a graph. Subjects were tested on the performance of these nine commands on sessions 2 to 6 (but using different materials), by being presented with manuscript pages compatible with each of the three environments, marked with editing instructions which required the use of the ’A set’ command sequences. The materials were similar to those which the subjects had used to practice the commands. This procedure resulted in five progressive samples of performance on each of the nine frequently used commands.

Anecdotal evidence and personal experience suggest that a division is made by users within the command set of a system, between the subset of commands which are used regularly and the ’other’ commands. Roberts and Moran (1983) have defined a set of core commands for text editing systems which encompass the functionality required to produce any real output. This set is comparable with the subset of system commands which most users regularly apply (e.g. ’delete text’, ’enter text’). In contrast, there are many commands available on computer systems which even experienced users do not use, and of whose details they are often uncertain. These may not be learned until some advanced task must be performed on a system. For example: a user may not use the commands to create a contents page or an index in a text processing system until producing a book, although considerable experience of the system may have been gained writing letters and memoranda.

The commands acquired early in system use will be learned at a stage when there is little knowledge of the system which could either aid in that learning or interfere with it. The commands acquired later during system use will be learned when there is a larger body of knowledge held about the system. This knowledge should benefit the later learning of commands if they are consistent with it, but may inhibit learning if they are incompatible.

This development in the range of commands used has been actively embodied in the ’training wheels’ approach to learning computer systems (Carroll and Carrithers, 1984) where users are initially exposed only to the core of commands required to fulfil their task. Users are exposed to other commands only after they have attained some proficiency with the core commands. This technique has been shown to lead subjects to perform basic tasks more quickly than subjects who use the whole system initially (Carroll and Carrithers, 1984; Catrambone and Carroll, 1987). This approach can be integrated with the initial view of the user outlined above through the ’chunking hypothesis’ and three assumptions which support it (after Rosenbloom and Newell, 1987):

The Chunking Hypothesis: A human acquires and organises knowledge of the environment by forming and storing expressions, called *chunks,* which are structured collections of the chunks existing at the time of learning.

Performance Assumption: The performance program of any system is coded in terms of high-level chunks, with the time to process a chunk being less than the time to process its constituent chunks.

Learning Assumption: Chunks are learned at a constant rate on average from the relevant patterns of stimuli and responses that occur in the specific environments experienced.

Task Structure Assumption: The probability of recurrence of an environmental pattern decreases as the pattern size increases.

This hypothesis suggests that the components of command sequences will be progressively clumped together into chunks until a whole command sequence is represented as a single chunk. The fewer chunks are required to perform a command sequence then the less time the sequence will take to perform. Consequently, commands learned at an early stage will have no or few relevant constituent chunks present and will require the development of a procedure for the sequence from the minimal parts of the sequence. The representation of commands learned later may include chunks which were developed for commands already learned. However, they will still require the development of some structures which were not previously defined and the appropriate recruitment of those which are. This process may be easier if the new commands conform to a characterisation of the commands already learned (Barnard et al, 1981). In contrast to chunks of performance sequences, such characterisations may take the form of high level rules governing the system command structure and operation which users may have abstracted. The performance tests were designed to capture this hypothesised variation between commands which are learned early in system experience and frequently used, and those learned later and used more rarely.

Whereas the ’A set’ tested core commands of the class described by Roberts and Moran (1983), the ’B set’ performance test was designed to assess the acquisition of new functions as learning progressed and to assess whether users incorporate new functions more easily as a result of the accrual of general system knowledge. To do this, command sequences learned in one session were tested in a subsequent session. However, in each case the command sequences were tested only once. In tests from sessions 4 to 6, 18 different command sequences were tested. The command sequences required to perform the ’B set’ tasks incorporated sections of the command sequences required for the ’A set’ tasks. For example, the command sequence for ’move text’ in the ’A set’ required the menu selection of ’move’ and then a selection of the area of text to be moved and the target point; the ’B set’ command sequence for ’copy text’ required the selection of the menu item ’copy’ and then the same selections to define the body of text and target site. In this way, the ’B set’ command sequences incorporated chunks which had already been established for ’A set’ commands. If the chunking hypothesis and its associated assumptions hold true for the learning of computer command sequences, then the ’B set’ commands should be learned more easily than those for the ’A set’ since chunks intermediate between the single actions and the full command sequence should already exist.

To decide if the knowledge acquired generalises to untaught functions a third performance test (the C set) was administered to capture expert performance on commands unknown to the expert. This test was administered at the end of session 6, and involved users attempting six tasks which required the use of novel system functions. The command sequences for these functions were not explained to the subjects, although they required the selection of menu items which appeared on menus with which subjects were familiar from their general system use. This test was intended to show the extent to which general system knowledge could be recruited to perform the high level task of discovering the appropriate command sequence for novel functions. However, the test was only attempted by the three fastest subjects, who made only five errors between them. This is too little data on which to answer the question conclusively, although this was a reasonable indication of the generalisation from knowledge of other parts of the system, for at least the fastest subjects.

The general model of learning described above does not characterise expert performance. Some psychological models of expertise (e.g. Card et al, 1983) have assumed that expert users make decisions as to the optimal method to perform, only on the basis of the time to complete a method. This expression may be too simple to explain performance and other factors may have to be taken into account. To assess if other factors are involved in the decision, the fourth performance test was designed. A data entry test was used to sample the choices made by experts between methods to achieve tasks. The data entry test was specifically focussed on the tradeoff between two methods of entering data into a spreadsheet (this study is reported in detail elsewhere: MacLean et al, 1985). This test took place in sessions 7 and 8. The two methods compared were 1) selecting each cell in a matrix using a mouse, and typing in numbers; 2) using a menu to establish an automatic cursor movement from cell to cell, then typing numbers into the cells of the matrix. The first method took longer for each cell, but the second included an overhead of a long setup time. Subjects used both methods for matrices of various sizes for one hour after their other experience with the system. Following this practice, they were asked to choose, and use, one method to enter data in each of 12 different matrices in both directions. For small matrixes the mouse method was faster; for large matrices the setup time for the menu method became proportionally smaller, so that method was faster. Since subjects used each method frequently before they were asked to choose the optimal method for each matrix, it was possible to calculate the time each subject took to perform each stage of the entry methods. Consequently, it was also possible to calculate at which matrix size the temporally optimal change in methods occurred for each subject.

For all tasks performed, a timestamped videolog was preserved for all user-system dialogue and user-experimenter interaction. For the first three tests, four different analyses were then performed: 1) the time subjects took to perform the task; 2) the number of tasks achieved without major re-attempts; 3) the number of tasks achieved in a single goal; 4) and an analysis of all deviations from an optimal route for performing the tasks (for the A set, these are reported in detail in Wilson et al, 1985a). In order to make such analyses, a task analysis and model had to be constructed to specify tasks, goals, major attempts and local deviations. The explicit model used for this study contained two parts. The first assumes that users have a top level goal to complete what is presented on the manuscript marked for editing. Within this they then have subgoals to complete each of the three tasks marked. Users then make attempts to achieve each subgoal by constructing and executing a method. The optimal method for achieving a sub-goal was taken to be that which users had been shown on videotape during training; attempts include sub-optimal sequences as well as this method. It is assumed in the model that at each stage in the execution of this attempt, subjects can test the consequences of their actions. If tests fail they have options: 1) to make a local correction to the attempt; 2) to construct a new attempt to complete the subgoal; 3) to postpone the present subgoal and attempt another. If all the tests in an attempt are passed then the goal is completed. This model was further specified to account for the exact nature of the local correction, re-attempts, and goal changes within the command language of the VisiOn system (this is presented in Wilson et al, 1985a and Wilson et al, 1985b and will not be described here in detail). In general this model divided an attempt into four stages: the specification of the attempt; the establishment of the context in which to issue the commands specific to the task; the performance of commands specific to the task; and a termination of the command issuing context. Actions were associated with movement between each of the states in the model so that a local correction to an attempt consisted of returning to a higher level menu and making another selection, using the delete key on the keyboard or similar actions which did not modify the nature of the entire attempt. The actions associated with the construction of a new attempt at the subgoal included an abort of the attempt to the top level menu for the environment and starting the task again. There is a special case where a goal is completed by the user, but the route prescribed was not used. These were classed as ’Ignored Failures’.

This formal model of the task permitted a very exact classification of the actions of users which yielded results which were quantifiable. This contrasts with the informality found when observers note their reactions to a user’s performance. Unsurprisingly, in the A set test the times to complete the overall tasks fell across sessions ($F(2,14) = 24.62$; $p < .001$) and the number of overall tasks completed without major errors increased ($F(4,28) = 3.72$; $p < .02$). These gross measures appear to support the assumption that general learning reduces performance time. However, a more exact analysis shows that the times to perform three tasks fell dramatically (e.g. the time to enter a formula into a spreadsheet fell from 245 to 80 seconds) while the times for the other six tasks remained constant. Similarly, a gross measure of the number of command sequences performed without major re-attempts rose significantly from 65% to 80% ($F(2,14) = 4.82$, $p<.03$), although for five of the nine commands, and three of the eight users they did not.

The relationship here between total time and the percentage of command sequences performed without major re-attempts is not straightforward. Two of the four commands whose performance involved a decrease in major re-attempts are also associated with large improvements in time, with the other two being associated with more modest improvements. In contrast, one command sequence which shows substantial gains in the time to complete it (the sequence for typing text into a word processor) was performed with a consistently low level of accuracy throughout.

This time/accuracy variation demands a more detailed analysis of the data which shows that there was also a change in the classes of errors across sessions. The proportion of local corrections while attempting tasks doubled (see Table 1); in contrast the proportion of major re-attempts dropped by two-thirds, while the proportion of errors due to changes to goals remained constant.

Test Session	Local Correction	Re-Attempts	Goal Changes	Ignored Failures	N
A set 2	39.7	39.7	17.6	3.0	68
6	67.8	12.9	16.1	3.2	31
B set 4	37.5	40.0	15.0	7.5	40
6	64.9	27.0	8.1	0.0	37

Table 1. Percentages of deviations from optimal route by class.

Overall, there are both users and tasks for which performance remains poor, and others for which it improves. It was also true that neither the time to carry out a correct command sequence nor the number of retries after a failed command reduced significantly. These results offer little evidence for the speeding up of performance for well formed sequences or their proceduralisation. Rather, the reduction in performance time is due to the reduction in major re-attempts.

It can be concluded that a view of the learning captured by this performance test which suggests that all sequences, or performance on all aspects of the system would improve synchronously, is inappropriate. Rather than a single progression, the data suggest that it would be more helpful to consider the gross performance measures as reflecting an averaging of individual command sequences each of which could lie at different stages of development within a user's repertoire of knowledge and skill. This repertoire as a whole develops, but parts of this knowledge are inaccurate, and these may persist and cause consistent errors.

The B set test was analysed in a similar fashion to the A set. Since the B set involved the testing of different command sequences of varying length on each session, it would be inappropriate to present time data. However, to answer the question as to whether the general improvement in system knowledge aided the acquisition of new command sequences, one can rely on error data. The number of errors for the sequences in the first B set test was constantly the same as that for the third A set test (see table 1), whereas the error count for the A set test fell by half over the four sessions. Despite a temptation to interpret the absolute number of errors as an indication that users are recruiting chunks of knowledge established during the A set test during their acquisition of B set command sequences, this would be inappropriate since the B set sequences were not perfectly balanced for length with those in the A set. In contrast, the constancy of the error count for B set tests could suggest that the acquisition of later command sequences was not being progressively aided. However, this too would be inappropriate since the items were not the same across the B set sessions, and this would effect the relative count. However, an analysis of the pattern of errors is valid evidence, and this indicates that the learning of the later B set commands is being aided. The shift in error types across sessions for the A set test that accompanied the reduction in performance time, was from major re-attempts to local corrections. The same shift exists in the errors for the B set from sessions 4 to 6. This change in error patterns can be attributed to users learning to recognise errors in time to avoid the need for major re-attempts. This suggests, that the information in the user's repertoire which is permitting them to move from time consuming major re-attempts on well practiced commands also affects the performance of the new sequences in the B set test.

Whereas the A and B set tests where designed to indicate the development of proceduralised methods as users approach a state of expert performance, the last performance measure was intended to test how choices were made between methods when a state of expertise has been reached. The data entry task was designed to assess whether time alone was the criterion which experts used to select the best method to achieve a task as is often assumed. The method allowed the calculation of the temporally optimal entry method for any data matrix for any subject. After much practice and experience, the users then selected the method they wished to use, and used it. The choice points actually used by subjects were not the same as the temporally optimal point but showed a consistent bias in favour of the method including the use of a menu to set the direction of cursor movement over the mouse method. One explanation for this could be that users judge the relative performance times for the two methods inaccurately. However, incidental descriptions by the users suggest that they were aware of the relative performance times. This simple study of a small set of expert performance illustrates that the assumption that expert decisions are based on time alone is inadequate to explain the actual choices made. Further factors have to be introduced into models to make accurate estimates. In this study, anecdotal evidence suggests that using both a mouse to select spreadsheet cells and a keyboard to enter numbers breaks the users' concentration or is mentally effortfull. Spending time to establish the direction of cell movement on a menu and then using a keyboard to enter numbers was considered easier. Consequently, factors of cognitive load, or focus of attention must be included in models of the criteria by which experts choose methods as well as performance time if they are to be accurate.

4. Tests of Articulatable Knowledge

The model held of the user in this paper incorporates a distinction between procedurally and declaratively represented knowledge. It has been assumed that procedural knowledge supports expert performance, but is not articulatable. Declarative knowledge exists in parallel with this which is articulatable. Consequently, this declarative knowledge should also be investigated as part of an investigation of the changes which take place during learning.

If users give verbal protocols during performance, one assumes that this protocol reflects both the way that users perform the present task, and the way they would perform that task outside the laboratory. Both of these assumptions have been challenged. It has been suggested that such techniques present a danger of the verbalisation distorting the normal operation of cognitive processes (e.g. Ericsson and Simon, 1984). Consequently, methods of filtering protocols have been developed (e.g. Bailey and Kay, 1987) or alternatives to the simple talk aloud protocol have been suggested, either to the form and volume of data collected (e.g. Hammond et al, 1983), or in the method of eliciting protocols through tasks other than the concurrent verbalisation of a single user performing a task (e.g. Suchman, 1985).

An attempt to combine these advantages in a task which probes users’ knowledge was made in the prompted knowledge elicitation (PKE) test in the study of VisiOn which would give access to the users’ articulatable knowledge, without imposing a concurrent protocol task. The purpose of this test was to assess the changes in verbalisable knowledge during learning, and to evaluate the relationship between this and the performance measures. In the PKE test users are presented with pictures of screens, about which they are asked focused questions (see Canter and Brenner, 1985). The picture based interviewing technique, reduces the quantity of protocol to be analysed by presenting pre-specified questions about specific sample system states. It also permits a formalisation and quantification of the analysis by having pre-specified sets of target claims which could be produced by users in reply to the questions. The formalisation of the task and dialogue, also permits a specification of the structure of the PKE task, and of its analysis. From this it is clear that users’ responses are not purely expressions of their knowledge of the system, since the sources of information which are likely to influence a protocol content include the knowledge of the domain of application, information in the photograph, the interpretation of the picture probe task itself, and the prompts from the experimenter, in addition to the knowledge of the system being studied.

A detailed report of this study is presented elsewhere (Barnard et al, 1986), so only a brief outline of the method and results will be presented here. The test was administered at the end of sessions 3 and 6, where the same 16[2] pairs of photographs were used. Each screen was presented in two different pictures. One where domain information was present and another where it was absent. Data will be collapsed over this condition in the present description. Users were given initial instructions which put them in the position of someone explaining the system to a friend. The experimenter prompts in this study, focused about: 1) the task the user would be performing to be in the state presented in the picture; 2) the command sequence that would have reached the screen portrayed in the picture; 3) the sequence from this screen that would be used to achieve the suggested task. Subjects verbal responses were recorded and transcribed. The responses to the probes can be regarded as providing several ’statistical’ samples of a user’s repertoire of system knowledge at the two points in the learning programme when the test was administered. Accordingly, a scoring procedure was devised to capture the content, form and attributes of that knowledge as

[2] Only data from the first 9 picture pairs is presented.

expressed in the verbal protocols. The scoring scheme divided the protocol into ’core propositions’ or ’claims’ that were identifiably different for a particular photograph. This procedure was similar to the approach adopted by Long et al (1983) with protocol data from groups discussing computer use. Claims made by the subjects which contained information that fell within classes defined by the probes were judged as target claims, while other claims were judged to be non-target and were dropped from the analysis. Within the classes of target information, user claims were scored as, true (accurately describing system operation); false (in contrast to system operation); inexact (a claim neither clearly true nor false); or indeterminate (for ambiguous claims and explicit statements of ignorance).

The analysis of the protocols showed a rise in the mean number of target claims from session 3 to session 6 (Wilcoxon T=3, p<.05) indicating that the users’ verbalisable knowledge was increasing. It is possible that this rise was caused by an increased familiarity with the test, or ease in talking to the experimenter, but there are few grounds for this explanation. This overall rise is due to an increase in the number of true claims (subjects: Wilcoxon T=3 p<.05; materials: Wilcoxon T=3.5, p<.05), since the number of false, inexact and indeterminate claims did not rise significantly. However, within this rise there is a complex change in the exact claims which were repeated at the two sessions (46% of true claims at session 6 were repeated; 31% of false claims).

The rise in true claims is not unexpected, and the fact that some false claims recurred is consistent with prior observations based on protocol examples and with the repertoire view of knowledge outlined above. However, two thirds of the false claims made in session 6 were new. Although these new claims arose for a number of reasons, more detailed analysis suggests that false claims were made in spite of known facts that were accurate. With one of the probes for example, only one user correctly identified it as the prompt state for editing cells in the spreadsheet. Although true claims were elicited concerning cells and work area actions, those concerning the task and prior menu selections nearly all focused on the entry rather than the editing of material. This confusion between menu paths was understandable in terms of their gross similarity. However, other probes elicited true claims from users which accurately described prompt states for entry. They had apparently not accessed that knowledge to block false inferences concerning the edit prompt. Thus, while the number of true claims may rise, the underlying knowledge from which the claims were derived may remain compartmentalised and functionally inaccessible in a novel context.

Having established some characteristics of the verbalisable knowledge acquired by users and changes in it during learning, it is necessary to investigate the role that that knowledge may play, and circumstances under which it may be recruited.

5. Tests Requiring Implicit use of Knowledge

There are a large number of possible off-line recall tests, however no strong relationship between recall and the usability of computer systems can be shown (MacLean et al, 1984). Despite this, if some evidence exists for a particular form of user representation being important in determining performance, other indirect measures can often provide useful converging evidence to form a more detailed conclusion. Two off-line tests were included in the VisiOn Study which were designed to elicit different aspects of users’ knowledge of the system and its operation. These were a questionnaire and a command name sorting task which were administered only in session 7.

The questionnaire included 40 questions to which users had to give true/false answers and confidence ratings on a four point scale. Specific questions were constructed in the light of the general user performance on previous tests. The purpose of the questionnaire was to test if users would recognise states of affairs or system rules even if they could not articulate them, and conversely, to see if they would endorse false claims that were consistent with common performance errors. The mean correct response rate was 70%, and of the 96 errors there were 73 false positives and 23 false negatives, with 11 questions accounting for 63% of the errors.

Of the questions which were consistently answered wrongly, some support user performance errors and others contradict them. One question asserted that the terminator ’done’ occurred on all command menus. This is false, and yet was affirmed by 5/8 subjects (with a mean confidence of 1.6; 1 = very sure, 4 = complete guess). The item ’done’ was responsible for a large number of performance difficulties in the A set test, but the performance problems were mostly due to users forgetting to select it when available, rather than attempting to select it when it was not available. Since the statement contradicts performance, the affirmation indicates uncertainty rather than the presence of false knowledge. This affirmation of a falsehood supports the idea that where there are performance difficulties, they are due to ambiguity or uncertainty in the subjects mind. In the light of this, it is surprising that users were extremely confident in the questionnaire that all their answers were correct (mean confidence = 1.35).

The most common user error in the A set test was associated with typing text into an editor. This is the only task for which no menu selection need be made - it is the default state. However, many users searched the menu hierarchy for a command which would allow them to do this. The only question which all users answered incorrectly (with a mean confidence of 1.9), stated that there was an option at a specified point which enabled the typing of text. This again is totally consistent with performance errors, however, all users managed to type text for each of the A set tests and on many other occasions although they never made a menu selection. Further, if they had a knowledge of the menu hierarchy at the point specified in the question (which they had frequent experience of) they would have ruled out the possible presence of a menu item there. These two examples both support the view that users are willing to endorse false statements about the system which are consistent with user errors. Such endorsements of false statements provide useful information about areas where errors in performance are likely to occur.

Both the PKE task and the questionnaire suggest that users are not good at recalling the menu structure in order to rule out interpretations, the name sorting task was designed to test incidental learning of the menu structures of the system. This data would supplement that for the C set performance data, and the PKE, to support a view of the representation that could support problem solving behaviour.

The VisiOn system contains three environments. Each environment uses a top level menu to select commands. Many of these menu selections result in the presentation of a second menu list. The ’cue’ component of the task was comprised of the main menu lines from each of the three environments. The list of stimulus words to be sorted under each of these headings were taken from the menus that arise when menu items on these lists are selected. A total of 72 secondary menu items were selected from four menu headings in each of the environments. Subjects were presented with a large sheet of paper on which were headings consisting of the main menu lines from the system. They were also given an alphabetic list of the 72 target words. They were instructed to write each target word under menu item that would be selected to get that word on a menu. Out of 576 items there were 314 assigned to some heading of these 141 were assigned correctly and 173 incorrectly. For the items that were used in the A set test, on average 6 subjects placed them correctly. For items used in the B set test, on average 3.3 subjects placed them correctly. Of the items not used, those that occurred on menus with the A set items were placed correctly by 1.6 subjects on average; those that occurred on menus with B set items were placed correctly by 0.8 subjects; and those that occurred on C set menu were placed correctly by 0.1 subjects.

This evidence suggests that users are developing some representation of the menu structure and that the accuracy of this representation corresponds to the frequency of usage of a particular menu. However, it would appear that there is very little representation of items that have not actually been used themselves. The conclusion that can be drawn from this test is that users do not represent an abstract menu hierarchy incorporating a rich representation of unused items which could be recruited to solve novel problems.

6. Interrelationship of the Tests

This paper has briefly described the results from six assessments of user knowledge which provided sufficient data to be analysable. There are three aspects of the interrelationships of these tests which must be considered. Firstly, how do the test results combine together? Secondly, do the results indicate important individual differences. The third aspect, concerning what the results from the tests as a whole indicate about the learning of computer systems will be discussed in the next section.

The initial view taken of user learning distinguished between verbalisable and non-verbalisable knowledge. Consequently, the PKE test was used to access verbalisable knowledge and performance tests were used to test non-verbalisable knowledge. If these tests access bodies of knowledge which are recruited at different times then there data should be incompatible. Users can accurately perform tasks in the A set, whereas they cannot either describe them in the PKE, or assign their menu items in the sorting task. This contrast supports the argument that performance and non-performance tests either access different knowledge, or access the same knowledge differently (e.g. Ericsson and Simon, 1984). The objective of this study was to investigate the representations established during learning, but for practical purposes only the representations which support performance may be of interest. Consequently, is there any evidence for the knowledge illustrated in the tests of verbalisable knowledge, or those involving an implicit use of knowledge supporting performance?

There are cases where the evidence from different tests coincide. However, these combinations of evidence result from interpretation, since the data themselves are of different forms. For example, common errors in the A set, are associated with false confirmations in the questionnaire; similarly, commonly used functions in the performance tests are associated with correctly placed menu items in the sorting task. In both these cases, there is confirmatory evidence from the tests requiring an implicit use of system knowledge for findings from the performance tests.

One measure of the overlap between tests, is to rank the eight subjects from the ’best’ to the ’worst’ on each of the measures used, and compare the extent to which the relative ranks overlap. The rankings for selected measures are presented in table 2. From this, there is a significant correlation between the PKE and the A set performance test (r_s = 0.85; p< 0.01), showing that users who displayed accurate performance also demonstrated more accurate verbalisable knowledge of the system than those whose performance was less accurate.

TABLE 2: Intersubject differences and variability. The A set performance test is reported by the percentage of tasks completed at the first attempt. The B set performance test, by the total number of errors made by subjects. The recall test, by the number of items correctly assigned. The questionnaire, by the number of questions correctly answered. The PKE test, by the number of true core claims produced. There are two measures derived from the data entry task: the time to complete simple manual tasks (a human simple speed measure) and a distance measure from the point of optimally efficient tradeoff for each subject (users 5 & 6 always chose one method, hence their tradeoff point is at infinity).

Subjects	Ranks and Ranges							Mean
	Aset	Bset	Recall	Quest	PKE	Data Entry		Rank
	Tasks	Number				Completion	T/O	
	1st attempt	Errors	(No)	(No)	(claims)	time (sec)	point	
1	5	8(61)	6	5	6	3	3	5.1
2	6	5	8(7)	6	7	4	2	5.4
3	4	4	4	3.5	4	8(83)	4	4.5
4	3	2	5	3.5	3	2	5	3.4
5	7	6	7	7	8(22)	7	7.5	6.7
6	8(51%)	7	3	8(23)	5	6	7.5	6.3
7	2	3	2	1(33)	2	1(54)	1(0.6)	1.7
8	1(98%)	1(7)	1(30)	2	1(90)	5	6	2.4

The highest correlation between tests is between the questionnaire and the A set test ($r = .97$, $p < 0.01$). The measure from the A set, is one of users' ability to construct or retrieve methods which will successfully achieve the task demands, and to perform these without major errors. This appears to contrast with the questionnaire where users confirmed or denied generalisations about system function. However, the questionnaire was designed to confirm observations made during the A set test so the content of the two is linked. However, this test does indicate that the types of knowledge tested for in the questionnaire could be those which support performance in the A set test. However, it is also possible that although they test different knowledge, subjects who are good at one test are good at the other.

The measure which is the worst correlate with the other appraisals is the speed of keying in the data entry task. However, since all the other measures sample learning, and the data entry task samples expertise on a predominantly manual task, this is not surprising. It is, however, interesting to note that user 8 shows good performance on the learning tasks while his performance on both aspects of this more expert task is poor. This contrasts with subject 7 whose performance on the learning tasks was also consistently good, but whose performance on the expert task was correspondingly good.

There is little in this small population sample that can be used to argue about individual differences. The experimenters observations of the users during the experiment suggested that user five had considerable difficulty with all the tasks, and also she is ranked last overall in table 2. The observed difficulty manifested itself in an unwillingness to act without reassurement. However, her rank is very close to that for subject 6, who did not show such obvious difficulties. It is tempting to argue that there could be three populations represented in table 2. Subjects 5 and 6, show the worst overall performance, subjects 7 and 8 show the best performance and the remaining four subjects yield average data. However, the sample is too small for such a conclusion to have more than face validity.

7. Summary of Indications Concerning User Learning.

The initial view taken of user learning was based on a three stage model (after Fitts, 1964 and Anderson, 1983) of a reduction in performance time and errors with task experience arising from the chunking of knowledge (after Rosenbloom and Newell, 1981) which results in a state of expert performance where the choices between methods to achieve tasks are made on the basis of performance time (after Card et al, 1983). Should the present data motivate any changes in this view?

The data from the A set test show no evidence of a simple reduction in performance time during learning. Instead, it appears that there are some tasks and some users for which there is improvement and others for which there is not. This cannot be used to argue against the initial view taken of learning, since it may be that the users in this study did not reach a stage of proceduralisation even for established tasks. In order to interpret these data it was suggested that user knowledge should be viewed as a repertoire of knowledge fragments which changes during learning, and which is sampled by the various tests.

The A and B sets sample different stages in the development of the repertoire. The changes in the error patterns for these two command sets follow the same pattern, despite one set were being consistently practiced while the other was not. This indicates that the change in error pattern was taking place over the sessions rather than over practice with the individual commands. This change was from the frequent occurrence of errors that lead to the performance of new attempts to perform tasks, to the dominance of errors which could be corrected locally. From this it would appear that the dominant change in knowledge was either, that which lead to the application of tests during attempt execution in order to identify errors, or the ability to recover from errors locally.

Both the A set and the off-line tests illustrate the development of user knowledge on core commands as learning progresses. The PKE test also showed that as learning progresses not only is false information removed from the repertoire, and that more true information is acquired, but new false information is also acquired.

The data from the sorting task add to this picture of a developing repertoire by suggesting that users had not developed a representation of the menu hierarchy even by the end of the experiment, but were performing methods in a sequential manner. Although, the users' knowledge of the menu hierarchy was better for frequently used commands than for those which could have been acquired incidentally, rules about the structure of the hierarchy were also being learned. For example, the questionnaire results showed that users affirmed rules that a terminator 'done' appearing on all menus, and all methods requiring an initial menu selection, even though they were false generalisations. These inaccurate rules appear to be applied in many of the attempts which result in user errors in the performance tests. Consequently, it would appear that errors in performance arise when there is a conflict between elements in the repertoire, leaving an ambiguity in the creation of attempts to perform tasks. This ambiguity is resolved by the application of a high level rule which may be inaccurate, resulting in an attempt which results in error when enacted.

From these data as a whole a picture has developed of a repertoire of knowledge which changes during learning. The repertoire contains fragments of performable methods, high level system rules and some structures representing commonly used commands. Some of these are accurate to the system, others are not. As learning progresses, some of the method fragments may combine, but they do not do so for all methods at the same rate. The most significant change is in fragments that permit recovery from error states which become more accessible when those states are encountered. There is also both a loss of false information but new false information is acquired, which may be contradictory to other knowledge fragments. When users have to perform a task by constructing an attempt and encounter a conflict they default to a system rule although it may be inaccurate.

8. Conclusion

No simple technique can be expected to adequately capture the complexity of using a computer system, although it may provide useful local information about a specific aspect of performance. However, a range of carefully related different measures can give sets of detailed information which provide converging evidence on which to base a more detailed conclusion. This paper has shown the application of such an approach in one study for which it has proved to be extremely successful in increasing understanding of how various aspects of the user interface affect both user performance and the underlying mental representation on which it is based.

The representation that appears to underly the learning captured by these tests is best described as a repertoire of knowledge fragments. This repertoire develops over the course of learning, and different measures and performance tasks sample different sets of this repertoire.

9. References

Anderson, J.R. (1983). *The Architecture of Cognition.* Mass.: Harvard University Press.

Bailey, W.A. and Kay, E.J. (1987). Structural Analysis of Verbal Data. In J.M. Carroll and P.P Tanner (eds.) *Proceedings of CHI+GI 1987 (Toronto, April 5-9).* New York: ACM.

Barnard, P.J., Hammond, N., Morton, J., Long, J. and Clark, I. (1981). Consistency and Compatibility in human-computer dialogue, *International Journal of Man-Machine Studies*, **15,** 87-134.

Barnard, P.J., Wilson, M.D., and MacLean, A. (1986). The elicitation of system knowledge by picture probes. In *Proceedings of CHI 1986.* New York: ACM.

Canter, D. and Brenner, M. (1985). *Uses of the Research Interview* London: Academic Press.

Card, S., Moran, T. and Newell, A. (1983). *The Psychology of Human-Computer Interaction.* Hillsdale, N.J.: Lawrence Erlbaum Associates.

Carroll, J.M. and Carrithers, C. (1984) Blocking learner error states in a training wheels system. *Human Factors,* **26** (4), 377-389.

Catrambone, R. and Carroll, J.M. (1987) Learning a word processing system with training wheels and guided exploration. In J.M. Carroll and P.P Tanner (eds.) *Proceedings of CHI+GI 1987 (Toronto, April 5-9).* New York: ACM.

Ericsson, K.A. and Simon, H.A. (1984) *Protocol Analysis, Verbal Reports as Data.* Cambridge, Mass.: MIT Press.

Fitts, P.M. (1964). Perceptual-motor skill learning. In A.W. Melton (ed.) *Categories of Human Learning.* New York: Academic Press.

Hammond, N., MacLean, A., Hinton, G., Long, J., Barnard, P.J. and Clark, I.A. (1983). *Novice use of interactive graph-plotting system.* IBM Hursley Human Factors Laboratory Technical Report HF083. Hursley, U.K.: IBM.

Lemmons, P.A. (1983) Guided Tour of VisiOn. *BYTE,* **8** (6), 256-278.

Long, J., Hammond, N., Barnard, P.J. and Morton, J. (1983). Introducing the interactive computer at work: The user's views. *Behaviour and Information Technology* 2, 39-106.

MacLean, A., Barnard, P.J. and Hammond, N. (1984). Recall as an indicant of performance in interactive systems. In B. Shackel (ed.) *INTERACT '84.* Amsterdam: North Holland.

MacLean, A., Barnard, P.J. and Wilson, M.D. (1985). Evaluating the human interface of a data entry system: user choice and performance measures yield different tradeoff functions. In P. Johnson and S. Cook. (eds.) *People and Computers: Designing the interface,* Cambridge, U.K.: Cambridge University Press.

Neisser, U., Novick, R. and Lazar, R. (1963) Searching for ten targets simultaneously. *Perceptual and Motor Skills,* **17,** 427-432.

Neves, D.M. and Anderson, J.R. (1981) Knowledge compilation: Mechanisms for the automatisation of cognitive skills. In J.R. Anderson (ed.) *Cognitive Skills and Their Acquisition.* Hillsdale, N.J.: Lawrence Erlbaum Associates.

Newell, A. and Rosenbloom, P.S. (1981) Mechanisms of skill acquisition and the law of practice. In J.R. Anderson (ed.) *Cognitive Skills and Their Acquisition.* Hillsdale, N.J.: Lawrence Erlbaum Associates.

Rosenbloom, P.S. and Newell, A. (1987). Learning by Chunking, a production system model of practice. In D. Klahr, P. Langley, R. Neches (eds.) *Production system models of learning and development,* Cambridge, Mass.: MIT Press.

Suchman, L.A. (1985). *Plans and Situated Actions.* Xerox Palo Alto Research Report, ISL-6. Palo Alto, Ca.: Xerox.

Snoddy, G.S. (1926) Learning and Stability. *Journal of Applied Psychology,* **10,** 1-36.

Wilson, M.D., Barnard, P.J. and MacLean, A. (1985a). *User Learning of Core Command Sequences in a Menu System.* IBM Hursley Human Factors Laboratory Technical Report, HF114, pgs 117. Hursley, U.K.: IBM.

Wilson, M.D., Barnard, P.J. and MacLean, A. (1985b). Analysing the Learning of command sequences in a menu system. In P. Johnson and S. Cook. (eds.) *People and Computers: Designing the interface,* Cambridge, U.K.: Cambridge University Press.