

Error Handling: Visualisations in the Human-Computer Interface and Exploratory Learning

Stephan Dutke

Freie Universität Berlin, Germany

La visualisation de l'information guidant l'utilisateur dans l'interface homme-ordinateur est devenue l'une des activités les plus recherchées dans l'exploitation de la bureautique et des systèmes informatisés. On étudie ici la relation entre les visualisations, l'exploration active du système et le traitement des erreurs de manipulation. On décrit, dans la visualisation de l'information guidant l'utilisateur, deux dimensions en interaction illustrées par un menu à déroulement: pour ce qui est de l'aspect quantitatif, le rapport des items du menu constamment affichés à ceux qui ne le sont pas peut varier; quant au versant qualitatif, c'est le contenu sémantique du nom du menu ou autres dénominations des rubriques qui peuvent être modifiés. Dans notre expérience, chacun des quatre groupes de débutants en informatique ($N = 29$) s'est familiarisé avec un éditeur de textes en cinq sessions grâce à de courtes instructions suivies d'une exploration autodirigée. Les deux dimensions, avec chacune deux modalités, ont servi de variables indépendantes (2×2 groupes expérimentaux). L'intensité de l'exploration active autodirigée du système devint une covariable. L'efficacité du traitement des erreurs de manipulation a été évaluée à travers l'analyse des stratégies de passage entre différentes phases d'action. La dimension qualitative de la visualisation a un effet faible mais positif sur le traitement des erreurs. Par contre, la dimension quantitative a un impact négatif, mais seulement indirectement, par l'intermédiaire de la motivation de l'utilisateur à explorer activement le système: une augmentation du rapport des items constamment visibles à ceux qui ne le sont pas conduit à une exploration moins intensive, ce qui aboutit à une moindre connaissance verbale du système. Une exploration superficielle risquait plus de générer des problèmes supplémentaires dans les situations d'erreur que d'aider à traiter efficacement les erreurs.

Visualising user information in the human-computer interface has become one of the most preferred design features in office automation and communication systems. In the present study, the relationship between visualisations,

Requests for reprints should be sent to Dr. Stephan Dutke, Institut für Psychologie, Freie Universität Berlin, Habelschwerdter Allee 45, D-14195 Berlin, Germany.

Thanks are due to Dieter Zapf, Jerilyn Lewter, and Wolfgang Schönpluf for their useful comments on an earlier version of this manuscript.

active system exploration, and error handling is investigated. Two interdependent dimensions in the visualisation of user information are described and exemplified by a pull-down menu system: (a) with regard to the quantitative aspect, the ratio of permanently visible to non-permanently visible menu items can vary; (b) with regard to the qualitative aspect, the semantic content of menu names or other high-level category descriptors can vary. In the present experiment, each of four groups of computer novices ($N = 29$) learned a text editor in five sessions by means of short instructions followed by self-directed exploration. Both dimensions, with two levels each, served as independent variables (2×2 experimental groups). The intensity of self-directed, active system exploration served as a covariate. Error-handling efficiency was evaluated through the analysis of transition patterns between different action categories. The qualitative dimension of visualisation was shown to have a weak but positive effect on error handling. In contrast, the quantitative dimension was found to have a negative relationship to error-handling efficiency, but only indirectly through users' motivation to explore the system actively: an increased ratio of permanently visible to non-permanently visible menu items led to less intensive exploration, which resulted in less verbal knowledge about the system. Less exploration was more likely to generate additional problems in error situations than to help handle errors efficiently.

VISUALISATIONS IN THE INTERFACE AND ERROR HANDLING

Because of the high mental and economic costs of errors during computer work, software designers strive for systems that diminish the probability of user errors and enhance the chance of effective error management. A preferred design strategy for application systems involves visualising user information in the human-computer interface. Visualisations of system states or command options are salient features of menus or direct manipulation interfaces (Shneiderman, 1982). In such cases, users are supported by pop-up or pull-down menus, icons, windows, dialogue boxes, and graphic indicators for different purposes.

Theoretically, visualisations in the interface can be conceptualised as an external memory store (Schönpflug, 1986; 1989) in that users are provided with information about the system which would otherwise have to be retrieved from memory. Command names, parameters, file names, or syntactic rules are represented in the interface itself. There are several advantages of visualisation which can be directly related to error handling. In the present context, error handling is considered to include any user-initiated problem-solving activity appropriate for the correction of former user errors or compensation for their unwelcome consequences.

First, whereas in human memory retrieval often fails, the external store is highly reliable. Consequently, skills acquired in systems offering rich visualisations are primarily recognition-based (Howes & Payne, 1991;

Mayes, Draper, McGregor, & Oatley, 1988) whereas in command-driven interfaces, more recall-based skills are acquired. As recall imposes higher demands on memory than does recognition, the probability of finding a solution to an error-recovery problem is generally expected to be higher in interfaces characterised by a high degree of visualisation than in those not manifesting high degrees of visualisation.

The second advantage is related to planning: a great portion of visualisations serve to inform the users of what types of actions they can execute in a given interactive state. This facilitates planning because the system displays the opportunities available in the current state, and the user may even be motivated to try an unknown opportunity. Moreover, rich visualisations allow for limited pre-planning (Payne, 1991) because the current problem state is displayed and therefore does not need to be extracted from a complete plan stored in the working memory (cf. Larkin, 1989). Therefore, error recovery may be improved, because planning how to eliminate or to compensate for the unwelcome consequences of former errors is facilitated.

Third, visualisations are designed to provide a graphic "environment" on the screen (Canter, Powell, Wishart, & Roderick, 1986). Because the visual appearance of this environment reflects different system states, the working memory load is reduced, and the user's orientation within the system should be enhanced. System orientation is an important prerequisite for effective error handling (Nievergelt, 1982).

Finally, visualisations allow for feedback consisting of spatial and movement elements. This type of feedback, as opposed to the distinct feedback generated in traditional command-driven interfaces, has been shown to enhance cognitive control (Te'eni, 1990). Cognitive control is enhanced because the richer feedback helps the user to understand "not only that a decision is a mistake, but also *why* it is a mistake" (Te'eni, 1990, p. 454). Such insights have a substantial effect on the promotion of effective error handling.

The most frequently investigated examples of visualisations are menus (Kreigh, Pesot, & Halcomb, 1990). In fact, many of the arguments mentioned earlier can also be applied to menu-driven interfaces: (a) menu selection is assumed to be primarily recognition-based; (b) it facilitates planning by presenting executable commands that vary with the current system state; (c) menus are part of the graphic environment presented on the screen to enhance orientation; and (d) the appearance or disappearance of a menu as a consequence of user operation provides direct feedback.

On the other hand, the menu example is well suited to demonstrate a problem common to all visualisations. A menu system consists of permanently visible menu items as well as non-permanently visible, context-dependent items. A simple and well-known example is the system of pull-

down menus. In such a system the commands are hierarchically arranged so that only top-level items (the menu names as shown in the menu bar) are displayed permanently. After selecting one item, only the contents of this menu are visible. It is impossible to use a command unless the user knows in which menu it is housed. Similarly, a command parameter represented in a dialogue box can only be changed if the user succeeds in making this dialogue box visible by activating the appropriate command. In other words, gaining access to non-permanent, context-dependent visualisations, such as commands or dialogue boxes, requires knowledge about the structure of the whole menu system, that is, knowledge about the structure of the external memory stores.

There are two basic dimensions in visualisations that are believed to affect the acquisition of this knowledge: first, the ratio of permanently visible to non-permanently visible menu items. If this ratio is increased (for instance, more permanently visible menu items associated with fewer non-permanently visible items), access to the remaining non-permanently visible items is facilitated. In terms of studies investigating the depth-breadth trade-off in menu systems (Lee & MacGregor, 1985; Miller, 1981) increasing the ratio corresponds to broadening the hierarchy. At the extreme, all commands may be displayed simultaneously and permanently. This would provide a survey of all available commands and an unhindered access (Kommers, 1988; Parton et al., 1985; Patrick & Fitzgibbon, 1988). Of course, this is not a realistic solution because of limited screen space—unless such a map is not represented in the interface itself, but externally, for example on a second screen (Widdel & Kaster, 1987) or in written learning materials (Fitzgibbon & Patrick, 1987).

If it is not possible to increase the ratio of permanently visible to non-permanently visible items, the second dimension becomes more crucial: the semantic content of high-level menu items (in the present example, the names of pull-down menus). All items that are not permanently visible (e.g. commands or dialogue boxes) can be accessed with the help of items on higher levels (such as menu names or commands). The permanently visible items then serve as category descriptors, provided that there are significant semantic relationships between the (non-permanently visible) target items within one menu and between the (permanently visible) descriptor and all target items within this menu (Kreigh et al., 1990; Snowberry, Parkinson, & Sisson, 1985). The lower the ratio of permanently visible to non-permanently visible items, the more precise these semantic relations must be in order to ensure an unhindered access.

This association between descriptors (menu names) and targets (commands and dialogue boxes) also provides a classification system for commands, which communicates similarities and dissimilarities among commands to the user (Kreigh et al., 1990). Thus, the menu names can

impart knowledge about the meaning of commands (Carroll, 1983). For instance, a command named "Copy" may be interpreted in different ways, depending on whether it is part of a menu named "Files" or of a menu named "Edit".

With regard to error handling, manipulations in both dimensions (e.g. raising the ratio of permanently visible to non-permanently visible items or enhancing the semantic content of permanently visible items) do not seem to be equivalent. The former manipulation enhances only command accessibility. In contrast, enhancing the semantic content of menu names not only facilitates command accessibility, but also may provide additional information about meaning and function of commands. Therefore, enhancing the semantic content of menu names is expected to support effective error handling more than will raising the ratio of permanently visible to non-permanently visible menu items.

EXPLORATORY LEARNING AND ERROR HANDLING

Exploratory learning in human-computer interaction is described as a self-directed activity (Briggs, 1990a) which results in the testing of subjective hypotheses about internal relationships within the system (Shrager & Klahr, 1983; 1986). In this perspective, exploratory learning is by no means aimless (Robert, 1987), provided that the learner acts on the basis of a small, but correct, body of pre-knowledge (Briggs, 1988, 1990a, 1990b; Schindler, 1987; Schindler & Schuster, 1990).

Exploratory learning has repeatedly been shown to improve the abilities contributing to successful error handling. Kamouri, Kamouri, and Smith (1986) compared persons who learned a technical device by free exploration with those who learned through written instructions. The exploring persons transferred knowledge more effectively to a new situation by using analogies. Studies by Greif and Janikowski (1987) and Frese et al. (1988) demonstrated that free-exploring learners acquired more correct knowledge about the system. Moreover, users whose system exploration is guided by individual preference are more likely to experience unknown and unintended system states. Therefore, they are assumed to be more familiar with unforeseen system states (Greif, 1986). This should have a positive impact on the user's competence to handle errors. Frese et al. (1991) compared a training procedure that deliberately provoked user errors and self-paced attempts to correct them with training based on error avoidance, and found positive effects for the error training. For these reasons, several attempts have been made to support exploratory learning, not only through active user training but also through the use of adequate software design strategies (Howes & Payne, 1990; Paul, 1992).

VISUALISATIONS AND EXPLORATORY LEARNING

In the preceding sections, two potential factors influencing error handling have been described: visualisations as a factor of the human-computer interface, and exploratory learning as a factor of training. It should be noted that these factors are related: visualisations not only affect error-handling performance directly, but may also affect the intensity and style of learning activities. If self-directed system exploration is actually centred around system knowledge acquisition, then its intensity should be inversely related to the degree of visualisation. The less system features are visualised, the greater the necessity to acquire this knowledge by active exploration. The validity of this hypothesis depends on three conditions: (a) the discrepancy between the user's preknowledge and the system's complexity is small enough to enable the user to explore the system in a goal-oriented manner; (b) the user is motivated to acquire knowledge about the system and is not hindered by the "motivational paradox of the active user" outlined by Carroll and Rosson (1987); and (c) there is a sufficient degree of freedom in the training procedure to allow individual exploration.

GOALS OF THE PRESENT STUDY

In the preceding sections, some of the interrelationships between visualisation, exploratory learning, and error recovery were discussed. It follows that error handling may depend on characteristics of visualisation strategies as well as on active system exploration. Concerning visualisation, two interdependent dimensions were described that represent a quantitative and a qualitative aspect of visualisations in general, and of pull-down menu systems specifically: (a) the ratio of permanently visible to non-permanently visible menu items, and (b) the semantic content of menu names. It is predicted (Hypothesis 1) that raising the semantic content of the menu names will be more effective at promoting error handling than will raising the ratio of permanently visible to non-permanently visible menu items. Hypothesis 2 predicts that error handling will also be positively affected by more intensive exploration. The relations predicted in Hypotheses 1 and 2 may be complicated by the assumption that less visualisation may stimulate the learners to explore the system more intensively (Hypothesis 3). In this case, a paradoxical and indirect effect of visualisation may be possible, in that less visualisation may lead to more or deeper exploration, which may improve error-handling capabilities. This means that the intensity of active system exploration may moderate the relationship between visualisation and error handling. Therefore, different degrees of visualisation are conceptualised as independent variables, intensity of exploration as a covariate, and error handling behaviour as a dependent variable.

With respect to experimental methods, the following two aspects were important considerations: first, most of the existing studies on the effects of visualisations suffer from several confounded factors. Often, totally different systems were compared (Altmann, 1987; Roberts & Moran, 1983; Whiteside, Jones, Levy, & Wixon, 1985) thus preventing differences in user performance from being solely attributable to different degrees or types of visualisation (Landauer, 1987; Svendsen, 1991). Comparisons of different interface types often treated information design and interaction technique as a joint factor (for example Rauterberg, 1989a). Investigating the benefits of visualisation requires variations in the interface design without any differences in system functionality or interaction technique. Therefore, the present study used variations of the same system that differed only with respect to the theoretically relevant dimensions (ratio of permanently visible to non-permanently visible menu items, and semantic content of the menu names).

Second, the investigation of error handling requires the analysis of action sequences, not exclusively of separate error occurrences. This type of analysis is necessary because in many cases the user takes a more complicated path to circumvent problems or to compensate for negative consequences of previous errors which might have accumulated from several unsuccessful attempts to recover. Consequently, for the present study a taxonomy of actions was used instead of a taxonomy of errors. This taxonomy comprised categories of erroneous actions as well as categories of actions that were correctly executed, efficiently planned, and performed to correct previous errors or to eliminate their unwelcome consequences. Effective error handling is more likely to be detectable in such sequences of action categories than when only relative frequencies are investigated. Information about action sequences can be extracted from the action-transition matrices described in the data analysis section.

METHOD

Experimental Conditions

The participants learned a commercial text editor designed in desktop style. It was a direct manipulation system employing a one-finger mouse. A total of 41 commands were available, organised in six pull-down menus with permanently visible menu names in the menu bar at the top of the screen.

There were two independent variables with two levels each. The first variable was the ratio of permanently visible to non-permanently visible menu items: half of the participants had a figure above the screen displaying all 41 commands and their corresponding six menus. The other half of

subjects had no such help; commands could be visualised only by opening a menu with the mouse. In both situations it was impossible to open more than one menu at one time. Thus, the menu figure provided a simultaneous overview of all of the commands that were otherwise not obtainable in this system.

The second independent variable was the semantic content of the menu names. One half of the subjects worked with the original system and its original menu names, such as "Data", "Searching", or "Symbols". The remaining subjects learned a version in which semantically neutral menu names (Digits 1 to 6) replaced the original names. Crossing both variables yielded four experimental groups: meaningful category descriptors as menu names with or without the menu figure, and neutral menu names with or without the menu figure. The four system versions were identical with respect to all other features. The intensity of demonstrated exploratory activity served as a covariate. There were repeated measures for the dependent variables (described later) as well as for the covariate.

Procedure

The instructions and tests were administered to each person individually. There were five experimental sessions of approximately two hours each, on five consecutive days in one week. All sessions, except the first and the last, consisted of three different phases: instruction, exploration, and testing.

Phase 1. *New Instructions*: in Sessions 1-4, the participants went through written materials explaining new editor functions. They were to execute each step with the system immediately after reading.

Phase 2. *Exploration*: after each instruction phase, the participants were allowed to explore all functions for 10 minutes without any constraints.

Phase 3. *Testing*: at the beginning of Sessions 2-5, the knowledge acquired in the preceding sessions was tested. Questions were asked concerning the properties of the system and its functions (knowledge of facts), and the sequence of operations necessary to attain certain goals (knowledge of actions). After answering the questionnaires, participants had to solve actual system tasks requiring the application of all the formerly instructed procedures. In Session 5, there was an additional transfer task, which required procedures that did not appear in any prior instruction.

Materials

Written instructions explained the system functions as well as the operating procedures. Four test tasks (Sessions 2, 3, 4, 5) and an additional transfer task (Session 5) were constructed. The test tasks required the application of all functions learned during the previous instruction phases. This means

that the requirements for accomplishing the test tasks accumulated across the four test phases. The tasks were written in colloquial German so that the subjects had to infer which editor functions were to be used to accomplish this task (for example: "Please print the letter named 'x' using the following new address: . . ."). The number of subactions necessary to complete a test task ranged from 6 to 24. In the example just described, the solution required three or four subactions (loading text, overwriting old address, or deleting old address and writing new one, and printing the document).

Verbal knowledge about the system was tested in two questionnaires. The first one, Knowledge of Facts, consisted of multiple-choice items concerning the meaning of commands (for example, "With the command 'Insert' you can (a) switch from overwrite to insert mode, (b) insert previously deleted characters, (c) insert a previously stored text"), and the allocation of commands to menus (for example, "In which menu do you find the command 'Insert'?"). The questionnaire Knowledge of Actions presented potential goal states and unarranged lists of relevant and irrelevant user operations (for example, "If you want to insert a sentence previously cut from the text then you have to (1) open menu 'Edit', (2) determine whether the sentence to be inserted is in the storage, (3) press the 'Insert-Key', (4) open menu 'Files', (5) click on 'Insert'"). The participants had to select the relevant operations and arrange them according to their correct sequence. There was a corresponding questionnaire item for each subaction required in the task of that session. Therefore, both questionnaires accumulated items across sessions in exactly the same manner as did the components of the test tasks, with items from former sessions recurring in subsequent sessions. Item order in the questionnaires was randomised in each session.

Segmenting the User's Stream of Behaviour

In order to evaluate how participants solved the test tasks, their behavioural sequences had to be segmented into meaningful, theoretically relevant units. Research from both theories of goal-directed behaviour (Oesterreich, 1981) and software ergonomics (Kieras & Polson, 1984) indicates that a state-transition representation is a useful concept for deriving such units. A state-transition net consists of a description of system states and the transitions between them. Such a representation can be constructed on different levels, depending on the definition chosen for transitions and states. As an initial step, we chose a level in which a transition is operationalised as an elementary user operation, for example, pressing a key or clicking once with the mouse. In this situation, the initial state-transition net represented the total amount of possible user opera-

tions. Whereas such a transition net adequately represents the functional properties of an interactive system, whether or not it provides a psychologically meaningful description of the problem space is questionable (Greif & Gediga, 1987). An individual's plan concerning how to solve a certain problem is probably represented more appropriately with sequences of subgoals and actions than with sequences of single keystrokes or mouse clicks. Therefore, the initial state-transition net was used only as a basis for deriving theoretically meaningful action units within the problem space of that system.

This process was performed in the following manner. In the initial state-transition net there were two categories of states, those that might represent goals or subgoals and those that might not. Potential goal states were defined in terms of the necessity for the user to decide at that point which system state should be worked towards next. In other words, a user operation might result either in a state leaving no other choice but to go on or return to the previous state, or it may result in a state that requires a decision and further planning. From this distinction, the definition of the fundamental unit of analysis was derived. We called the sum of the operations leading to a potential goal state a "subaction". The goal represented by this state is a potential subgoal which is part of the accomplishment of a more complex task. In the simplest case, a subaction consisted of exactly one user operation, but only if the achieved state was a potential goal state in accordance with the definition given earlier. In the system used in this experiment, a subaction usually consisted of two to four elementary operations. In other words, the user's behaviour was not described and evaluated in terms of elementary operations (for example, clicking on "Insert"), but in terms of subactions composed of several keystrokes or mouse clicks performed in order to achieve a certain goal (for example, the subaction "insert previously stored characters into a text" was composed of positioning the cursor, opening the menu "Edit", and clicking on "Insert"). Subactions were used as the fundamental units of analysis of a course of action during task processing.

Dependent Variables and the Covariate

The editor to be learned was run on an IBM-compatible personal computer with two screens, one for the participant and another for the experimenter. The experimenter's screen, which was not visible to the participants, mirrored all user operations, and was videotaped. Through this procedure, user behaviour was observed and recorded during both the exploration and test phases, but not during the instruction phases.

For each exploration phase for each subject we recorded how often menus were opened and how many different functions were spontaneously tried out. A score called Intensity of Exploration was computed by adding

both of these variables, and this measure served as a covariate. Thus, this score represented the aspect of breadth of exploration as well as depth.

The quality of task solutions was operationalised as the percentage of subgoals correctly achieved in each task. These subgoals were derived from a hypothetical optimal solution. This variable did not take into account which individual goal structures were generated by subjects; it is concentrated only on the results of task processing.

In order to evaluate how each task was processed, the course of action registered on the video was segmented into a sequence of subactions based on the previously described procedure. Each subaction identified on the basis of the initial state-transition net was categorised according to six classes of subactions:

1. *Correct subactions* were executed correctly, and their consequences facilitated the achievement of the overall task goal.
2. *Execution errors* were subactions that were executed incorrectly, although they were planned correctly, and did not help to approach the overall task goal.
3. *Corrected repetitions* were subactions that became necessary to correct a former execution error. The formerly incorrect subaction was then repeated with no errors.
4. *Planning errors* were subactions executed correctly, but the consequences were inappropriate for achieving the overall goal. Instead, planning errors often created additional negative consequences (for example, destroying previous work).
5. *Restoring subactions* eliminated or compensated for negative consequences of former errors. In contrast to a corrected repetition, a restoring subaction was not a repetition of the faulty subaction performed previously, but a totally different subaction. If, for example, a text still needed for further processing was erroneously deleted, it was not sufficient to repeat the delete subaction (now deleting the correct text), the erroneously deleted text must also be restored. Thus, a restoring subaction helped to recover from both execution and planning errors.
6. *Unnecessary subactions* had neither positive nor negative effects on the achievement of the task goal.

These six categories (inter-rater reliability = 0.89) were used to represent the complete individual action sequences present for each test task.

In the questionnaires, the errors per item were counted. For the multiple-choice items (Knowledge of Facts questionnaire), errors were considered to be either falsely accepted or falsely rejected alternatives. As there were three alternatives per item, with items having varying numbers of correct alternatives, there could be more than one error per item. The same held true for the items in the Knowledge of Actions questionnaire.

The following errors could occur in each item: omitting an indispensable operation; including a harmful operation; or changing the sequential order of two indispensable operations.

Subjects

The participants were 29 computer novices, ages ranged from 17 to 53 years with a mean age of 26. The 14 female and 15 male subjects had previously passed different vocational training programs ranging from academic to technical and administration qualifications. They were randomly assigned to the four experimental groups. All were paid for their participation. There were no group differences in age or two measures of intelligence (analogical reasoning and visual memory as measured by the WILDE-Intelligence-Test, Jäger & Althoff, 1983).

DATA ANALYSIS AND RESULTS

According to the experimental design, the percentage of achieved subgoals per task and the relative frequencies of subactions in the six action categories per task were analysed by means of ANCOVAs (2×2 independent variables, four repeated measures with intensity of exploration as a covariate). As outlined earlier, the analyses of frequencies of action and/or error categories were not expected to be helpful in the investigation of error handling without also considering sequential information about the course of action. Nevertheless, these results are reported, mainly to ensure that there were no substantial group differences in overall performance. Only in one case was there a marginally significant main effect of the independent variable Menu Figure ($F = 4.23$; $P < 0.1$): those users learning without the menu figure seemed to execute relatively more restoring subactions. There were no other effects of the independent variables, apart from several effects of the repeated measures factor, which indicate that task difficulty steadily increased from session 1 to session 5.

To investigate error handling more thoroughly it is necessary to take information about the course of action into account. The most basic information about action sequences could be provided through the analysis of the probability of transition from one action category to another. Therefore, each person's action sequence when performing each task was transformed into a complete transition matrix containing all six categories per person and per task. Then, the five matrices per person (five tasks) were averaged across all tasks, resulting in a single 6×6 action-transition matrix per subject. The question of interest was whether or not the structure of these person-specific action-transition matrices varied with the experimental treatment. In order to investigate this question, the number of variables in each matrix (36 possible transitions) had to be reduced. In

a first step, transitions involving correct and unnecessary subactions were eliminated from the matrices because their variances were found to be very low. The remaining 16 transitions, which involved execution errors, planning errors, restoring subactions, and corrected repetitions, represented the core information about error-handling sequences. In the second step, the reduced matrices were subjected to a VARIMAX rotated principal components analysis using the observed cell frequencies as input variables. Five factors with eigenvalues greater than 1, which together explained 74% of the total variance, were extracted. The first two factors could clearly be interpreted in terms of error-handling efficiency. Their composition is shown in Table 1.

The transitions loading high on Factor 1 can be interpreted as generating additional problems rather than handling errors effectively. Planning errors were followed by additional planning errors or execution errors, and execution errors were followed by new planning errors. Of course, there were also attempts at error correction, but these subactions themselves were followed by new planning errors with transitions from restoring subactions to planning errors demonstrating the highest factor loading. In contrast, Factor 2 is characterised only by adequate error-handling action transitions, without sequences involving additional planning errors.

In order to investigate the effects of the experimental design on the structure of action transitions, each person's scores on both factors were computed via regression analyses. Then, two ANCOVAs (2×2 independent variables and intensity of exploration as a covariate) were computed with the scores on Factors 1 and 2. This analysis revealed patterns differing substantially among both factors (Table 2). Using the scores on

TABLE 1
Factor Analysis of the Reduced Action-Transition Matrices:
Transitions Loading on the First Two Factors

Factor 1 ("Problem generation")		Factor 2 ("Efficient Error Handling")	
Transitions	Loadings	Transitions	Loadings
PE → PE	0.84	ExE → ReS	0.90
PE → Exe	0.71	Res → CoR	0.88
PE → ReS	0.77		
PE → CoR	0.68		
ExE → PE	0.70		
ReS → PE	0.90		
CoR → PE	0.69		

PE = planning error, ExE = execution error, ReS = restoring subaction, CoR = corrected repetition.

TABLE 2
Analysis of Covariance: Scores on Factors 1 and 2
(Extracted from the Reduced Action-Transition Matrices)

Dependent Variables	Source of Variation ¹	F ²
Scores on Factor 1	Exploration	8.10***
Scores on Factor 2	M-Names ³	3.23*

¹M-Names = independent variable "Menu Names", Exploration = covariate "Intensity of Exploration".

²* $P < 0.1$; *** $P < 0.01$.

³Means in Fig. 1.

Factor 1 (Problem Generation) as a dependent variable, results indicated a strong negative correlation with the covariate, Intensity of Exploration ($r = -0.52$, $P < 0.01$). No other effect was found. In contrast, there was no correlation between Factor 2 (Efficient Error Handling) and the covariate, but a marginally significant main effect of the treatment factor Menu Names—specifically persons learning with semantically meaningful menu names seemed to score higher on Factor 2 than did participants learning with semantically neutral menu names (Fig. 1). The first result clearly supported Hypothesis 2: The Intensity of Exploration was negatively correlated with the tendency to generate further problems when correcting former errors. The second result supported Hypothesis 1: Only one dimension of visualisation (semantic content of the menu names) had a direct positive effect on error handling.

In Hypothesis 3 it was assumed that the intensity of exploration itself may depend on the amount and content of visualisations in the human-computer interface. This assumption is even more important when considering the fact that exploratory activity had a strong impact on error handling (Hypothesis 2). To investigate whether or not exploratory activities differed among the four experimental groups, the variable Intensity of Exploration was subjected to an ANOVA (2×2 independent variables, four repeated measures). Figure 2 shows that persons learning without the menu figure explored more actively than did subjects learning with the menu figure, and this difference increased from Sessions 1 to 4.

This result is accompanied by a parallel finding concerning verbal knowledge about the system. Errors in the Knowledge of Facts questionnaire were standardised for the number of items in each session. An ANOVA with repeated measures revealed an interaction effect of the independent variable Menu Figure and the repeated measures factor. Both of the groups learning with the menu figure performed worse than the other two groups,

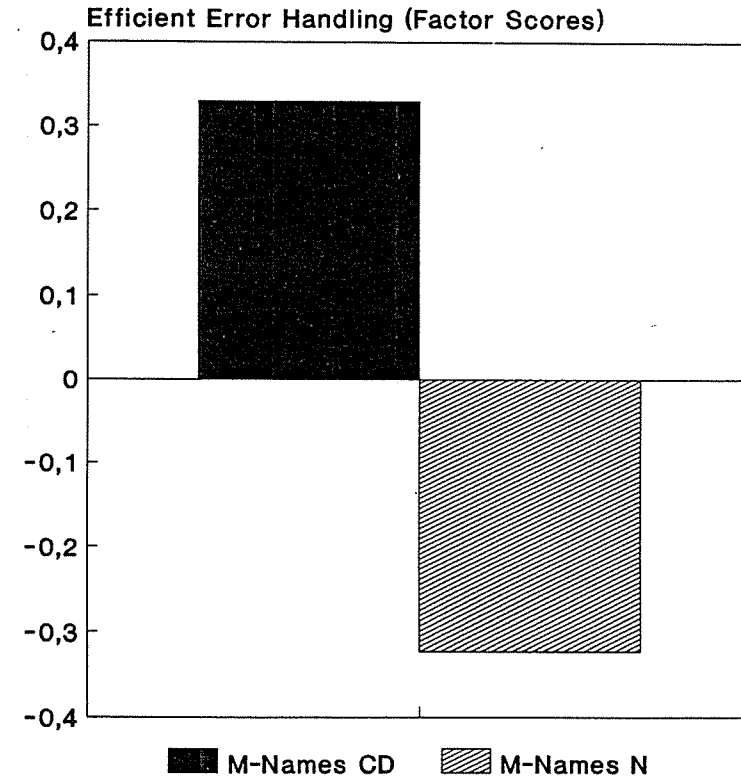


FIG. 1. Mean scores on Factor 2 ("Efficient Error Handling"). M-Names CD = menu names consisting of category descriptors, M-Names N = menu names consisting of numbers.

although the former started with fewer errors in the first test (Fig. 3). The relationship between visualisation (factor Menu Figure), exploratory activity, and knowledge about the system is in line with Hypothesis 3.

DISCUSSION

The theoretical rationale for the present study was the assumption of two different dimensions in visualising user information in the interface: the ratio of permanent to non-permanent visualisations, and the semantic content of visualisations. As exemplified by a pull-down menu system, it was predicted that changing the ratio of permanently visible to non-permanently visible menu items should not affect error-handling behaviour in the same manner as would manipulating the semantic content of high-level descriptors (such as menu names). The results reported here supported this assumption. This was apparent when comparing the factors Efficient

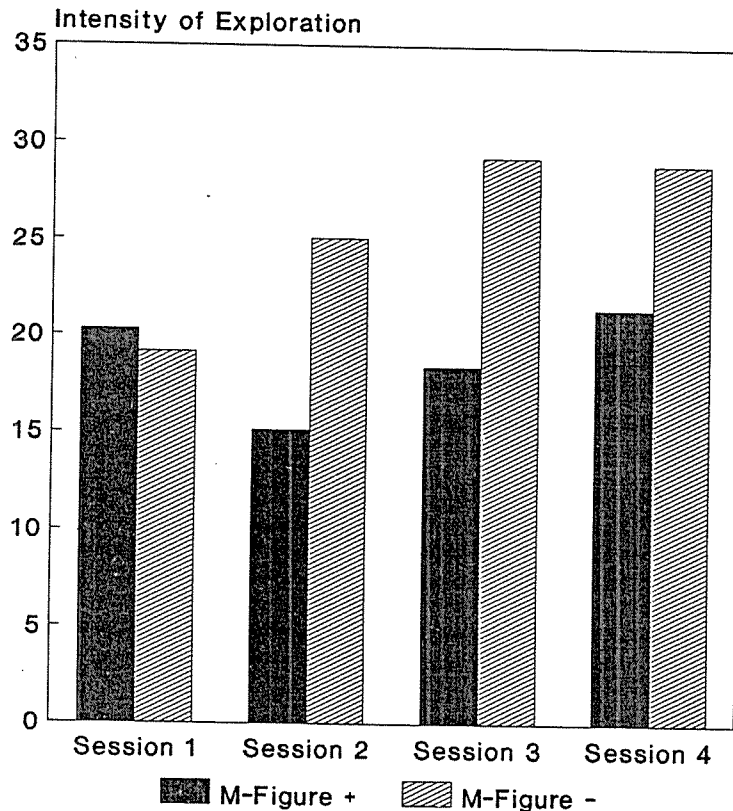


FIG. 2. Mean intensity of exploration (Sessions 1, 2, 3, and 4) for participants learning with menu figure (M-Figure+) and without (M-Figure-). Main effect factor M-Figure ($F = 6.05$; $P < 0.05$). Interaction effect M-Figure \times Session ($F = 3.56$; $P < 0.05$).

Error Handling and Problem Generation, which were found in the action-transition matrices. The scores on the factor Efficient Error Handling seemed to be affected by the independent variable Menu Names. Participants learning with meaningful menu names handled errors more efficiently than those working with semantically neutral menu names. This finding supported Hypothesis 1 and also the assumption that menu names served not only as a kind of "signpost" (Benest & Potok, 1984), but also to communicate functional features of the designated commands. Such category descriptors helped novice users to discover the meaning of commands, which improved their ability to eliminate or to compensate for the consequences of former errors. Moreover, this effect could not be explained in terms of different base rates of error occurrence. No differences in the relative frequencies of errors between the experimental groups were found.

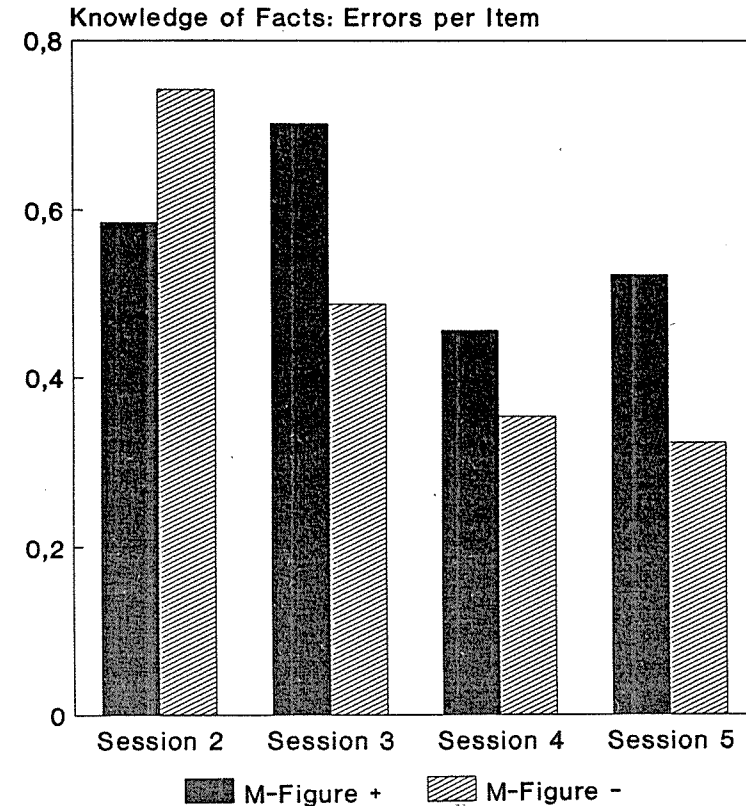


FIG. 3. Mean number of errors per item in the Knowledge of Facts questionnaire (Sessions 2, 3, 4, and 5) for participants learning with menu figure (M-Figure+) and without (M-Figure-). Interaction effect M-Figure \times Session ($F = 4.22$; $P < 0.01$).

The effect was exclusively due to the structure of action transitions, not to the absolute or relative number of errors.

Of course, one might question the sensibility of interpreting such a statistically weak effect, especially because of the non-standard use of the analysis of covariance (Evans & Anastasio, 1968). The reason for such an interpretation did not lie in an overestimation of the strength of the statistical effect, but in the fact that the other factor (Problem Generation) extracted from the reduced action-transition matrices showed a completely different pattern. The factor Problem Generation was not directly controlled by the independent variables: the effects of the experimental treatment were totally absorbed by the covariate, Intensity of Exploration. Participants who explored less had higher scores on Problem Generation (Hypothesis 2). But intensive exploration was more likely to occur in

groups learning without the menu figure. If a lesser amount of permanently visible user information was available (no menu figure), participants explored more intensively and divergently (Hypothesis 3). Additionally, they performed better in a test of verbal knowledge about the system. This pattern of results might be due to a levels-of-processing effect: intensively exploring persons not only compensated for missing information, but they also acquired more relevant knowledge than the groups not provided with a menu figure. This pattern is in line with studies of exploratory learning in human-computer interaction (Frese et al., 1988; Greif & Janikowski, 1987; Kamouri et al., 1986). Obviously, the participants' experiences from active learning could not simply be substituted for by the additional presentation of user information in the human-computer interface. On the other hand, motivation to explore the system actively seemed to have been diminished by the additional visualisation in the menu figure.

Of course, the present results should not be interpreted as a principal argument against visualisations. Whether they enhance or reduce motivation to explore the system depends on the specific relationship between the system's demands and the learner's capabilities. From research on curiosity, it can be concluded that a balance between system complexity and learner capability may provide for the most user motivation (Rauterberg, 1989b). In contrast, a predominance of system complexity may reduce exploration due to a resulting fear of losing control (Frese, 1987), whereas a prevalence of learner capabilities may also reduce exploration because of boredom and lack of challenge. In line with this reasoning, the additional visualisations in this experiment might have reduced the impression of system complexity thus causing subjectively low demands which resulted in a lack of challenge. Ironically, the menu figure, by creating the impression of an "easy to handle" system, prevented knowledge acquisition without compensating for memory deficits.

Manuscript received April 1992

Revised manuscript received January 1994

REFERENCES

- Altmann, A. (1987). Direkte Manipulation: Empirische Befunde zum Einfluß der Benutzeroberfläche auf die Erlernbarkeit von Textsystemen. *Zeitschrift für Arbeits- und Organisationspsychologie*, 31, 108-114.
- Benest, J.D., & Potok, M.H.N. (1984). Wayfinding: An approach using signposting techniques. *Behaviour and Information Technology*, 3, 99-107.
- Briggs, P. (1988). What we know and what we need to know: The user model versus the user's model in human-computer interaction. *Behaviour and Information Technology*, 7, 431-442.
- Briggs, P. (1990a). Do they know what they're doing? An evaluation of word-processor users' implicit and explicit task-relevant knowledge, and its role in self-directed learning. *International Journal of Man-Machine Studies*, 32, 385-398.
- Briggs, P. (1990b). The role of the user model in learning as an internally and externally directed activity. In D. Ackermann & M.J. Tauber (Eds.), *Mental models and human-computer interaction I* (pp. 195-208). Amsterdam: North-Holland.
- Canter, D., Powell, J., Wishart, J., & Roderick, C. (1986). User navigation in complex database systems. *Behaviour and Information Technology*, 5, 249-257.
- Carroll, J.M. (1983). Nameheads. *Cognitive Science*, 7, 121-153.
- Carroll, J.M., & Rosson, M.B. (1987). Paradox of the active user. In J.M. Carroll (Ed.), *Interfacing thought: Cognitive aspects of human-computer interaction* (pp. 80-111). Cambridge, MA: MIT Press.
- Evans, S.H., & Anastasio, E.J. (1968). Misuse of analysis of covariance when treatment effect and covariate are confounded. *Psychological Bulletin*, 69, 225-234.
- Fitzgibbon, L., & Patrick, J. (1987). The use of structural displays to facilitate learning. In H.-J. Bullinger & B. Shackel (Eds.), *Human-computer interaction—INTERACT '87* (pp. 611-616). Amsterdam: North-Holland.
- Frese, M. (1987). A theory of control and complexity: Implications for software design and integration of computer systems into the work place. In M. Frese, E. Ulich, & W. Dzida (Eds.), *Psychological issues of human-computer interaction in the work place* (pp. 313-337). Amsterdam: North-Holland.
- Frese, M., Albrecht, K., Altmann, A., Lang, J., Papstein, P.v., Peyerl, R., Prümper, J., Schulte-Göcking, H., Wankmüller, I., & Wendel, R. (1988). The effects of an active development of the mental model in the training process: Experimental results in a word processing system. *Behaviour and Information Technology*, 7, 295-304.
- Frese, M., Brodbeck, F.C., Heinbokel, T., Mooser, C., Schleiffenbaum, E., & Thiemann, P. (1991). Errors in training computer skills: On the positive function of errors. *Human-Computer-Interaction*, 6, 77-93.
- Greif, S. (1986). Neue Kommunikationstechnologien—Entlastung oder mehr Streß? Beschreibung eines "Computer-Trainings zur Streßimmunisierung". In K.-H. Pullig, U. Schäkel, & J. Scholz (Eds.), *Streß im Unternehmen* (pp. 178-200). Hamburg: Windmühle.
- Greif, S., & Gediga, G. (1987). A critique and empirical investigation of the "one-best-way-models" in human-computer interaction. In M. Frese, W. Dzida, & E. Ulich (Eds.), *Psychological issues of human-computer interaction in the work place* (pp. 357-377). Amsterdam: North-Holland.
- Greif, S., & Janikowski, A. (1987). Aktives Lernen durch systematische Fehlerexploration oder programmiertes Lernen durch Tutorials? *Zeitschrift für Arbeits- und Organisationspsychologie*, 31, 94-99.
- Howes, A., & Payne, S.J. (1990). Supporting exploratory learning. In D. Diaper, D. Gilmore, G. Cockton, & B. Shackel (Eds.), *Human-computer interaction—INTERACT '90* (pp. 881-885). Amsterdam: North-Holland.
- Howes, A., & Payne, S.J. (1991). Display-based competence: Towards user models for menu-driven interfaces. *International Journal of Man-Machine Studies*, 33, 637-655.
- Jäger, A.O., & Althoff, K. (1983). *Der Wilde-Intelligenz-Test (WIT). Ein Strukturdiagnostikum*. Göttingen: Hogrefe.
- Kamouri, A.L., Kamouri, J., & Smith, K.H. (1986). Training by exploration: Facilitating the transfer of procedural knowledge through analogical reasoning. *Journal of Man-Machine Studies*, 24, 171-192.
- Kieras, D.E., & Polson, P.G. (1984). A generalized transition network representation for interactive systems. In A. Janda (Ed.), *Human factors in computing systems. Proceedings of the CHI '83 Conference, Boston, 1983* (pp. 103-106). Amsterdam: North-Holland.
- Kommers, P.A.M. (1988). Textvision: Elicitation and acquisition of conceptual knowledge by graphic representation and multiwindowing. In G.C. van der Veer & G. Mulder (Eds.), *Human-computer interaction: Psychonomic aspects* (pp. 237-249). Berlin: Springer.

- Kreigh, R.J., Pesot, J.F., & Halcomb, C.G. (1990). An evaluation of look-ahead help fields on various types of menu hierarchies. *International Journal of Man-Machine Studies*, 32, 649-661.
- Landauer, T.K. (1987). Relations between cognitive psychology and computer system design. In J.M. Carroll (Ed.), *Interfacing thought: Cognitive aspects of human-computer interaction* (pp. 1-25). Cambridge, MA: MIT Press.
- Larkin, J.H. (1989). Display-based problem solving. In D. Klahr & K. Kotovsky (Eds.), *Complex information processing* (pp. 319-341). Hillsdale, NJ: Lawrence Erlbaum Associates Inc.
- Lee, E., & MacGregor, J. (1985). Minimizing user search time in menu retrieval systems. *Human Factors*, 27, 157-162.
- Mayes, J.T., Draper, S.W., McGregor, A.M., & Oatley, K. (1988). Information flow in a user interface: The effect of experience and context on the recall of MacWrite screens. In D.M. Jones & R. Winder (Eds.), *People and computer IV* (pp. 275-289). Cambridge, UK: Cambridge University Press.
- Miller, D.P. (1981). The depth-breadth trade-off in hierarchical computer menus. In R. Sugarman (Ed.), *Proceedings of the 25th Annual Meeting of the Human Factors Society*, (pp.296-300). Santa Monica, CA: Human Factors Society.
- Nievergelt, J. (1982). Errors in dialog design and how to avoid them. In J. Nievergelt, G. Coray, J.D. Nicoud, & A.C. Shaw (Eds.), *Document preparation systems: A collection of survey articles* (pp. 265-274). Amsterdam: North-Holland.
- Oesterreich, R. (1981). *Handlungsregulation und Kontrolle*. München: Urban & Schwarzenberg.
- Parton, D., Huffman, K., Pridgen, P., Norman, K., & Shneiderman, B. (1985). Learning a menu selection tree: Training methods compared. *Behaviour and Information Technology*, 4, 81-91.
- Patrick, J., & Fitzgibbon, L. (1988). Structural displays as learning aids. *International Journal of Man-Machine Studies*, 28, 625-635.
- Paul, H. (1992). EXPLORE—Interface building beyond the interface. In P. Brödner & W. Karwowski (Eds.), *Ergonomics of hybrid automated systems—III. Proceedings of the 3rd International Conference on Human Aspects of Advanced Manufacturing and Hybrid Automation* (pp. 295-300). Amsterdam: Elsevier.
- Payne, S.J. (1991). Display-based action at the user interface. *International Journal of Man-Machine Studies*, 35, 275-289.
- Rauterberg, M. (1989a). Ein empirischer Vergleich einer desktop- mit einer menuorientierten Benutzeroberfläche für ein relationales DBMS. In M. Paul (Ed.), *GI-19. Jahrestagung I. Computergestützter Arbeitsplatz, 18.-20. Oktober 1989* (pp. 243-258). Berlin: Springer.
- Rauterberg, M. (1989b). Über das Phänomen: "Information". In B. Becker (Ed.), *Zur Terminologie der Kognitionsforschung. Arbeitspapiere der GMD* (V1. 385, pp. 219-241). Sankt Augustin, Germany: GMD.
- Robert, J.-M. (1987). Learning a computer system by unassisted exploration. In H.-J. Bullinger & B. Shackel (Eds.), *Human-computer interaction—INTERACT '87* (pp. 651-656). Amsterdam: North-Holland.
- Roberts, T.L., & Moran, T.P. (1983). The evaluation of text editors: Methodology and empirical results. *Communications of the ACM*, 26, 265-283.
- Schindler, R. (1987). Preknowledge and the efficiency of self-guided user's learning. *Zeitschrift für Psychologie, Supplement 9*, 74-81.
- Schindler, R., & Schuster, A. (1990). On the relationship between a user's self-teaching and his knowledge. In D. Ackermann & M.J. Tauber (Eds.), *Mental models and human-computer interaction 1* (pp. 209-228). Amsterdam: North-Holland.

- Schönpflug, W. (1986). The trade-off between internal and external information storage. *Journal of Memory and Language*, 25, 657-675.
- Schönpflug, W. (1989). How shall a file be called? In F. Klix, N.A. Streitz, Y. Waern, & H. Wandke (Eds.), *Man-computer interaction research MACINTER II* (pp. 119-124). Amsterdam: North-Holland.
- Shneiderman, B. (1982). The future of interactive systems and the emergence of direct manipulation. *Behaviour and Information Technology*, 1, 237-256.
- Shrager, J., & Klahr, D. (1983). Learning in an instructionless environment: Observation and analysis. In A. Janda (Ed.), *Human factors in computing systems. Proceedings of the CHI '83 Conference, Boston* (pp. 226-229). Amsterdam: North-Holland.
- Shrager, J., & Klahr, D. (1986). Instructionless learning about a complex device: The paradigm and observations. *International Journal of Man-Machine Studies*, 25, 153-189.
- Snowberry, K., Parkinson, S., & Sisson, N. (1985). Effects of help fields on navigating through hierarchical menu structures. *International Journal of Man-Machine Studies*, 22, 479-491.
- Svendsen, G.B. (1991). The influence of interface style on problem solving. *International Journal of Man-Machine Studies*, 35, 379-397.
- Te'eni, D. (1990). Direct manipulation as a source of cognitive feedback: A human-computer experiment with a judgement task. *International Journal of Man-Machine Studies*, 33, 453-466.
- Whiteside, J., Jones, S., Levy, P., & Wixon, P. (1985). User performance with command menu and iconic interfaces. In L. Borman & B. Curtis (Eds.), *Human factors in computing systems II. Proceedings of the CHI '85 Conference, San Francisco* (pp. 185-191). Amsterdam: North-Holland.
- Widdel, H., & Kaster, J. (1987). Wirkungen visuell präsentierter Dialog-Strukturen auf die Interaktion ungeübter Benutzer mit dem Rechner. In W. Schönpflug & M. Wittstock (Eds.), *Software-Ergonomie '87. Nützen Informationssysteme dem Benutzer?* (pp. 329-339). Stuttgart: Teubner.