25

3. The Treatment of Errors in Learning and Training

Michael Frese and Alexandra Altmann

ABSTRACT

Since empirical research on errors in human–computer interaction is still in its beginnings, a theoretically derived taxonomy of errors is suggested in this paper. This taxonomy is based on action theory, differentiating levels of regulation and steps in the action process. Furthermore, slips and mistakes are to be distinguished. It is useful to analyse errors with the help of this taxonomy, for example whether errors stem from problems in the knowledge base for regulation, from goal and plan development and decisions, from lack of monitoring, or difficulties of perceiving or interpreting feedback. This is shown when qualitatively analysing different errors over the course of training and between different software systems (one on direct manipulation and one more traditional). We suggest that strategies of error management should be incorporated into the training process. Error management implies that one has to learn where errors will appear and how to deal with them effectively.

INTRODUCTION

We would like to discuss some functions of errors in learning and how one should deal with them in training computer-related skills. Essentially these are theoretical arguments but we also refer to one study of learning two different word-processing systems in which we recorded errors made by the trainees. This qualitative study serves mainly illustrative purposes. Our general notion is that the emphasis should be on error management. This means that training should teach the trainees to be able to deal with errors rather than to try to avoid errors. This should be so because people at work will invariably commit errors because not all the necessary features can be taught in training and new tasks and new software features appear. More-

Developing Skills with Information Technology
Edited by Lisanne Bainbridge and S. Antonio Ruiz Quintanilla

over, most software systems are not completely without problems, which may contribute to errors. If users have learned to manage errors effectively this will help them to be able to explore the system and to use errors as an inspiration to learn rather than to feel stressed. Before we can develop the notion of error management in training, it is necessary to introduce some general ideas on errors, including a taxonomy of errors.

ON THEORY

There are essentially two general lines of argument with regard to errors in training. One position argues that errors should be avoided as much as possible. This view has been particularly popular in two different scientific traditions, in behaviourism and in one particular brand of humanism. Alternatively, errors should be included into the learning process because they can also have a positive function. This position is more akin to cognitivistic concepts and to action theory.

What are the arguments for each position? The behavioristic tradition (particularly Skinner, 1953) argues that a person learns best through positive reinforcement. Errors are conceptualized to be punishment. Punishment does not lead to positive learning. Punishment just leads to a temporary suppression of a certain behaviour; it leads to emotional arousal and it does not tell the learner what he or she should really do (just doing nothing helps to avoid punishment as well) (Skinner, 1953). This line of argument led, of course, to the now famous programmed learning machines (Skinner, 1968).

There is a brand of humanism that proposes a similar way of thinking (although it uses a different nomenclature). Here errors are conceptualized to frustrate the student. Frustration leads to anxieties. Hence, the minimization of errors leads to a reduction of frustration and reduces the anxieties associated with learning.

Finally, behaviouristic quarters (less so Skinner than Hull and Guthrie) have argued that when an error is committed, one cannot help but learn something wrong. For example, every movement made leads to learning this movement. Therefore, this movement is repeated under certain circumstances even if the movement is known to be incorrect. Therefore, if one learns the correct movement immediately there are no competing response tendencies.

In contrast to this, cognitivistic and action theories argue that errors help under certain circumstances to actually increase the knowledge (mental model) about a system (for example Semmer and Pfäfflin, 1978). Thus, training should not attempt to restrict the chances to make errors but should incorporate 'typical' errors, train for them and use them to understand what one is doing. This latter position is, of course, related to some forms of exploratory learning (Bruner, 1960), since exploratory learning also uses learning from mistakes.

The above-mentioned differences in general educational approaches also exist in the literature on training human–computer interaction skills. In one reseach project, the system did not allow the trainees to make any errors at all (Carroll and Carrithers, 1984; Carroll and Kay, 1985). In sharp contrast, another study used error training (Greif, 1986). Both reported good results with their respective methods.

It is one of the tasks of this article to reconcile these different findings and different approaches to errors in the training process; to do this, we first want to develop a concept of errors and then discuss a theoretically derived taxonomy of errors.

The concept of errors

Errors produce the non-attainment of a goal. Not every non-attainment of the goal is an error, however. Only when actions have been performed that were potentially avoidable and that violated some rule can one speak of errors (Wingert, 1985). Every error violates some kind of goal (or supergoal) of the individual. As Norman (1984) pointed out, there are two types of errors: slips which result from wrong plans but right intentions and mistakes in which the intentions were wrong but the plan conformed to the intention. If a person deletes a file accidentally by hitting the wrong keys, it is a slip. If a person intends to delete a file, but recognizes later on that it is actually needed again, a mistake has been made. Note that even in the case of mistakes, the error is related to some higher-order goal. This higher-order goal may be, for example, not to waste time and energy (for example by having to write something twice).

This differentiation is not without problems. As the above examples show, behavioural data alone never tell us whether it was a slip or a mistake (in both cases the behaviour was the same—erasing the file). Even when we know the intention, however, it is not quite clear what a certain behaviour signifies. One could argue that the person who made the mistake actually made a slip in goal-setting (the wrong goal of erasing the file was set, even though it was still needed). In spite of these conceptual problems, the differentiation between slips and mistakes is phenomenologically useful (and particularly useful in the field of training). When a slip is pointed out, the person will most likely say, 'Oh yes, of course, it was an error.' If a mistake is pointed out, however, the person will want to be convinced because it is not obvious to the person that in fact a mistake has been made (because the fact of still needing access to the erased file is not in that person's working memory).

Mistakes usually signify some lack of knowledge or something missing in working memory. An additional characteristic of mistakes is that the boundary lines between mistakes and inefficient behaviours (Semmer and Frese, 1985) are blurred. People are, in general, oriented to behaving efficiently (Schönpflug, 1985). When a person makes a large detour, it will usually be called a mistake because the 'wrong' route was taken at some intersection. Was it a mistake or simply inefficient behaviour? It is hard to decide and we have to live with this conceptual fuzziness. If the person has the (super-)goal of not wanting to waste energy, making a detour was a mistake. If no such goal exists, it was inefficient behaviour. The judgement of whether a certain behaviour is inefficient can be done without having to refer to the person's goals. On the other hand, whether or not something is an error can only be judged by knowing the person's goals. Practically, the differentiation is not quite so important, since one of the tasks of training is to teach the trainee to make few 'detours' and hence few mistakes.

A taxonomy of errors

In Figure 1, we suggest a taxonomy of errors adapted from Frese and Peters (1988) which results from an action theory perspective (cf. Frese and Sabini, 1985; Norman, 1986). The steps of the action process consist of the development of goals and decisions between them, the development of plans and decisions, the execution of the plan and its monitoring, perception of feedback and interpretation of feedback. Influencing all of these steps is the knowledge base for regulation, an equivalent to the concept of an internal model. The knowledge base provides the material from which to develop goals and plans and to interpret feedback.

Similarly to Hacker (1973) and Rasmussen (1985), we distinguish four levels of regulation (after Semmer and Frese, 1985). These levels of the postulated hierarchy (or better heterarchy) can be distinguished by their generality (higher levels) and specificity (lower levels) and by whether they involve conscious thought. Actions regulated by higher levels require conscious attention; those regulated by lower levels are relatively automatic, with higher levels used only for occasional monitoring.

Sensorimotor level

This is the lowest level of regulation in which stereotypical and automatic movement sequences are organized without conscious attention. These highly automatic actions can be consciously regulated only within narrow limits because processing of feedback is done on the lower levels as well. Thus, substantial modifications are not possible at this level. Conscious regulation cannot modify such action programs; at most it can stop performance.

Steps in the action process

Levels of regulation	Goals/goal decisions	Plans/plan decisions	Monitoring execution of action	Perception and interpretation of feedback
Level of abstract thinking	N	N		N
Intellectual level of regulation	N	N	N / E	N
Level of flexible action patterns	N	N / E	E	N / E
Sensorimotor level of regulation		E	E	E
	Knowledge base for regulation			

Figure 1. A preliminary theoretically derived taxonomy of action errors. The knowledge base for regulation has an impact on all the steps in the action process; however, it has more influence on the upper levels of regulations. Black cells do not exist. N = novice; E = expert

Thus, action programs on the sensorimotor level may assert themselves—contrary to insight—because they have been well routinized. This is one reason for occasional slips, for example when a person wants to go shopping and drives a part of the way along a familar route that goes to the shop and to home. After a while the person finds himself at home without having done any shopping.

Level of flexible action patterns

At this level general action patterns are regulated that are relatively constant in their structure but whose parameters can be changed flexibly. Using block commands may be an example.

Intellectual level

At this level, complex analyses of situations and of problems are regulated. Learning a word-processing system is done at first on this level.

Level of abstract thinking

This is the highest level. General and abstract thought processes are regulated here; logical inconsistencies are tested and abstract heuristics are generated here. These heuristics may take the form of action styles, that is general and abstract heuristics that help in developing plans and goals and the use of feedback. The use of these heuristics may become automatized as well (Frese, Stewart and Hannover, 1987).

On the vertical side of Figure 1, the levels of regulation are described; on the horizontal side the steps in the action process are shown. The knowledge base for regulation has an influence on all of these steps of the action process.

Knowledge base for regulation

The knowledge base is the material from which a person can develop goals and plans, compare the action with a set of parameters and perceive and interpret feedback. Errors may appear here because of objective information deficits or wrongly conceived information and because of inadequate mental models or metaphors (including not knowing the boundary conditions of these metaphors).

Goal development and goal decisions

Errors may appear as inadequate development of goals, wrong decisions between goals or wrong derivation of subgoals. An example for an inadequate goal is an unrealistic goal and an example related to wrong decisions between goals may be unresolved goal conflicts.

Plan development and plan decisions

There is a large overlap between this step and goal development decisions. Errors can occur because of inadequate plans, wrong decisions between plans or derivation of inadequate subplans. Thus, errors can be due to inadequate or unrealistic methods and means. One error condition is due to inflexible use of plans or to giving up the plans before one can be really sure that they have not worked (Volpert, 1974).

Monitoring of execution of action

Sometimes it is possible to detect that one is about to make an error before one actually commits it. Monitoring helps here to reduce the error rate. The less one monitors one's actions, the more easily errors can occur.

Perception and interpretation of feedback

Here there are three conditions that lead to errors: not noticing feedback signals, wrong perception or wrong identification of feedback, and wrong interpretation of feedback signals.

Novices make mistakes more often and their mistakes are regulated in the upper part of Figure 1 (and have been labelled 'N'). Experts make more slips/errors because they are more often related to an automatic plan (or interpretation) of taking over an action; therefore experts show a preponderance of lower-level errors (they are presented as 'E' in Figure 1).

QUALITATIVE OBSERVATION OF ERRORS IN THE LEARNING PROCESS

This taxonomy is useful when looking empirically at errors in the learning process. The course of errors in the learning process and the different kinds of errors as a result of different systems could be observed during a long-term study (7 × 2 hours) of 24 students learning two different text-processing systems: the WordStar with its command language and the MacWrite (using the Macintosh) with its icons, its desk top metaphor and the use of a mouse (Altmann, 1987; Schulte-Göcking, 1987). The subjects had no experience with a computer. The training simulated the learning situation of a person who had just bought a computer and had to learn with the help of a manual; additionally the trainees used material and worked on problems that had been especially designed for this study. The experimenters observed the subjects at all times, intervening only after they had been asked for help and after the subjects tried out a first attempt to solve the problems themselves. Records on all errors were kept. We were especially interested in whether the errors followed a specific course over time.

Errors in the course of learning

The level of regulation as in Figure 1 is changed by learning. The novice regulates most actions on a higher level; after some practice the regulation is delegated to lower levels. This implies that the novice is most likely to make errors of the mistake kind. These mistakes may be due to knowing too little about the system, to using wrong or inadequate mental models or metaphors, to developing unrealistic goals and plans, to not knowing the right feedback signals and to misinterpreting the signals given. However, there is also room for slips that appear because attention is not given to the action at hand. Inattention leads then to leaving out, repeating or misordering crucial steps in an action sequence (Norman, 1981). The problem of inattention to important system parameters is crucial in the beginning of the training process, because there is constant information overload, since all of the information has to be consciously stored in working memory and little information is chunked.

The chances to make slips are greater for experts, however, because skilled behaviours contain many automatized components. Thus, they tend to regulate many actions on lower levels. Therefore, the lower levels may become activated without conscious control, leading to actions that were not intended—the most famous source of this kind of slip being the capture error (Norman, 1981).

Indeed, in the first sessions of the learning process, the subjects in our study had problems on all levels of regulation and with all steps of the action process. There was a high prevalence of mistakes, most of them due to the lack of basic understanding of word processing with a computer.

More concretely, special features of both systems like RETURN and BLANKs produced difficulties for the novices since they used a typewriter metaphor and, therefore, had problems understanding the special implications of these features in word processing. For example, they often used the RETURN key or the space bar instead of using the cursor keys, thus 'tearing apart' the text. Apparently, the knowledge base for regulation had not included a clear differentiation between cursor movement and editing functions.

After making errors such as the one above, the subjects had difficulties recovering from the error situation. Since they did not know that simply using delete commands would fix the text, the real trouble began now. The subjects were not able to interpret the outcome of their actions properly and, therefore, could not find the right problem-solving strategies. They only saw the text as 'torn apart' and were now searching for a command to 'combine', 'move' or 'format' texts. Since they did not know that RETURNs and BLANKs are represented and treated as characters in a word-processing system they could not find the solution. Moreover, these characters were invisible and, therefore, the subjects thought (and argued) that 'one cannot delete "nothing"'.

There is a difference between passively acquired knowledge and knowledge that is actively used. Purely 'telling' somebody the right answer did not help the subjects to actually use this information. Although they were told the abstract notion on how BLANKs and RETURNs are represented in a word-processing system and how to treat them, the subjects had to get to know the boundaries of their metaphor by actually making many mistakes before they could realize the uniqueness of these features in word-processing programs. Thus, it is sometimes necessary to make the errors actively before learning takes place.

The typewriter analogy did not only lead to errors on the intellectual level of regulation but also on the sensorimotor level. Thus, skilful typists had some particular problems in using automatized movements like pressing the RETURN key at the end of each line. Working with a word-processing system one has to use a RETURN key only at the end of each paragraph. Otherwise it is impossible to reformat the text at a later time. Again, there is a discrepancy between knowing something and being able to use the information. After a skill has been automatized, like pressing the RETURN key after each line, the reintellectualization process is difficult. Thus, this error was repeated over and over although the subjects knew that it was wrong. The habit first had to be broken to be able to establish new strategies of action.

Besides error prevention, error diagnosis (feedback perception and interpretation) is a special problem for the novice. For example, the CTRL key was sometimes not simultaneously pressed with the letter key so that the system did not interpret this as a command but as a letter. Thus a letter appeared on the screen that was not noticed by the subjects because they expected something 'big' to happen. Therefore, they thought that they had to find another command instead of the correct one just used.

Another typical problem in error diagnosis appears when the CTRL key is accidentally pressed instead of the shift key, thus giving a command instead of a capital letter. This error is also made by many experts, but in contrast to the novices they can rectify this relatively quickly. The novices became quite helpless because of the unexpected consequences of this slip (for example the appearance of the help menu or the next page on the screen). Since they could not interpret the feedback correctly, they were unable to find a solution to the situation.

In summary, in the first sessions of the learning process errors were made on all levels of regulation but the most prominent errors were made on the intellectual level of regulation, for example in the inadequate interpretation of feedback. Furthermore, an inadequate knowledge base was an important cause of errors.

However, as soon as the basic difficulties were solved, other kinds of errors dominated in later sessions. Even though comprehension errors were constantly made since the trainees learned something new in each session, slips and memory lapses became more important. Thus, the ratio of slips to mistakes changed, with slips and lapses becoming more frequent. These lower-level slips presented themselves as lapses in WordStar and as motoric coordination slips in MacWrite.

In WordStar, commands were forgotten or mixed up more frequently. For example, many errors using the command language system consisted of mixing up similar commands (like CTRL-KD with CTRL-KP or .op with .pa). Such lapses are the result of the abstact and confusing command names and an additional interference stemming from learning many new and different names. The reader may note that this area is not represented in the theoretical taxonomy of errors given in Figure 1. The study showed us that the important area of memory lapses should be incorporated.

When using the direct manipulation system, an example of those errors, which are often caused by the lower levels of regulation of Figure 1, is as follows: some subjects pulled down the menu but clicked at the wrong place because they were already thinking of the next step. Apparently, the subjects now felt more sure of their use of the commands, thus putting less attention on these matters and using the level of flexible action patterns. On the other hand, their level of expertise was not high enough, so that a higher degree of conscious attention would have been warranted. Thus, errors of the following type appeared: the subjects actually knew the correct procedure but made errors that they could recover from relatively quickly.

Moreover, in later sessions, the differences between the two systems led to different types of errors. Therefore,we will treat these differences in the next section.

Errors and different styles of human–computer interaction: direct manipulation versus traditional system

As mentioned earlier, two word-processing systems with different styles of human–computer interaction were learned in the experiment. WordStar is operated by a complex command language and offers additional menus as memory aids. This is called a traditional system. MacWrite is icon-oriented and presents a 'model world' with diverse objects that can be directly manipulated with the help of a mouse. The model world consists of a desk top metaphor with files, rulers or a waste paper basket. These two different interaction styles produced different kinds of errors in later sessions starting with about the fourth session.

In WordStar the menus are not self-explanatory and many subjects had difficulties using them. For example, it often happened that a person got into a submenu (for example CTRL-K) to search for a specific command (for example CTRL-KD). As soon as the correct command was found CTRL-KD was repeated without realizing he or she was in the submenu and, therefore, had to give only the second part of the command (for example D). The system, however, interpreted the input of CTRL-KD in the K submenu as CTRL-KK and, therefore, inserted a print control character for 'end of block' into the text. Most often, the person did not notice this, which could cause

great confusion later on. For example, the person might want to mark the beginning of a block further down in the text and would then receive an error message on the fact that the block marker 'end of block' was set before the one 'beginning of block'. This could not be interpreted since at this time the causal relationship of this error message to the prior action was no longer obvious to the learner.

Thus, the subjects did not grasp the hierarchical structure of the WordStar menu system—a problem of the intellectual level of regulation of the action plans. Moreover, the feedback on the errors was not clear and could not be easily interpreted. The result: error management was usually difficult. This led in turn to an overly cautious and non-exploratory strategy when using WordStar because errors were seen as a constant 'bother' rather than as an impetus to learn.

In contrast to WordStar, MacWrite did not present any major problems for using the menus or remembering the commands. Here the menus were very simple and clearly arranged and could be popped up quickly and without complication using the mouse. The errors using the MacWrite system were of a different nature and came about because of difficulties of the model world symbols and metaphors. There were three special problems:

1. Some procedures were too easy. For instance, some people had problems in activating a 'window'. In order to do this, one must simply point at the specific window and press the mouse key. The subjects expected a more complicated procedure. When they did not find any solution in spite of an intensive search, they resigned and asked the experimenter for help. Conclusion: the right procedure was *too* simple in comparison to their expectations.
2. Metaphors were taken too literally. For example, an object to be worked with must be marked at first by 'pointing *at* it'. Initially many people took this 'pointing *at*' too literally and moved the cursor that had the form of an arrow next to the object. Thus, they made the arrow point *at* the object instead of directly moving the arrow *onto* the object, which is the proper way of doing it.
3. Metaphors were designed inconsistently. For example, the function of the rulers in MacWrite is to set tabs, margins and spacing. This is not consistent with the actual function that rulers would have on a real-life desk, such as measuring something or drawing a line with them. Moreover, it is possible to make them invisible by using the command 'hide rulers'. This inconsistency of the design of the ruler symbols was the source of many errors. For example, the commands 'show rulers' and 'add rulers' were constantly mixed up, or the symbols on the ruler (for example for tabs) were falsely interpreted (for example as the bell ringing at the end of the line).

There is a clear difference in the number of errors of the WordStar group to the MacWrite group. More errors were made in the WordStar group because the subjects had forgotten commands or procedures and could not understand them from looking at the menus. Unfortunately we were not able to count the exact numbers of errors made with both systems, but our qualitative analysis tells us that approximately twice as many errors were made using WordStar.

When comparing the quality of the errors made with WordStar or with MacWrite it becomes obvious that they have completely different consequences. Errors using MacWrite very rarely caused a perplexing chaos as they did for WordStar. Because of this, MacWrite errors had few demotivating and frustrating effects. MacWrite supported error management much better.

THE FUNCTION OF ERRORS IN THE LEARNING PROCESS AND THE NOTION OF ERROR MANAGEMENT IN TRAINING

Our discussion so far suggests some hypotheses about the function and problems of errors in the learning process. Errors can have potentially positive as well as negative functions in the training process. This also gives us a clue as to what error management should mean. Error management strategies should be taught in training so as to enhance the positive functions of errors and minimize the negative functions.

The positive functions of errors in the training process

The positive function of errors conforms to the adage that one learns from errors. In one study a training group that received a sequential training (the subjects learned each command sequentially without any higher-order explanation) in which no errors were possible, performed worse than another group that allowed and encouraged the subjects to make errors (Frese *et al.*, in preparation). Errors provide feedback to the person. However, feedback is only useful when the trainee is able to perceive and interpret the feedback (an error message like 'error 024' will not do) and to leave the 'error situation' after having committed the error. Sometimes trainees get hopelessly lost when they find themselves in a situation that they did not want to be in and that they do not know how to change. Depending upon the level of regulation, there are different functions of errors. On the level of abstract thinking, errors help the individual to learn which metacognitions work and which ones do not. When the trainee learns that every 'typo' has quite negative effects when giving commands to a computer system, he or she will use the metacognition of meticulous typing (including checking before one 'sends off' the command with the return key). Similarly, when the person has learned strategies and heuristics to deal with errors effectively, the metacognition of using errors as challenges rather than as stressors will prevail. On the intellectual level of regulation, the boundary conditions of metaphors (cf. Carroll and Thomas, 1982) are discovered. For example, if the trainee thinks of the system in terms of a typewriter model, the boundary lines for this thinking become clear when he or she makes the error to write over a blank in the insert mode and finds out that this is not possible.

Errors usually lead to a reintellectualization of action patterns that used to be regulated on lower levels. When one employs a plan as usual (without thinking about it) and it suddenly does not work any more or produces error messages, one is forced to think about the action again. Therefore, errors (at least those with clear feedback) have the function of stopping premature automatization of a skill. Mistakes signify that the implications of a command may not have been completely understood yet and that one has to rethink and possibly retrain again. One prerequisite is, of course, that the training is done under realistic conditions where those errors that are prone to appear when performing real-life tasks will appear as well. There are many reports in the literature on sensorimotor learning (cf., for example, Volpert, 1981) showing that non-realistic feedback appearing concurrently with an action is worse than none because realistic errors can not be made under these conditions.

There is one more (positive) function of errors that is often overlooked: errors may spur creative solutions and new exploratory strategies. If the trainee does not know the difference between the 'insert mode' and the 'overwrite mode' and makes a mistake because of this, this might lead him or her to explore these different modes. Similarly, all the possible uses of an UNDO button (or back-up files) may be explored after one has made a mistake. Often, an accidental use of a command may produce an interest to find out how this command can be used and what its functions are.

This positive function can also be substantiated in our study on the learning process. When a menu was accidentally popped up using the MacWrite system this often led to an interest to explore the commands of this menu. This was so because the menus were presented very clearly and it was very easy to undo accidental errors. It often happened that a person said: 'Oh, what's this? I'd like to try it out!' The menus were therefore adequately used as a tool that stimulated exploration. Error management was supported by the system.

Additionally, subjects who made certain errors of basic understanding (that is those concerning RETURNs and BLANKs) earlier in the learning process were able to rectify them and to proceed without any further impediments. In each case, the other subjects would encounter this particular problem later on anyhow, but by this time they were already working on more complex problem situations, so that it was much more difficult for them to understand the actual error causes and to manage these errors effectively.

The negative function of errors in the training process

Given this set of advantages just discussed, one wonders why errors have a 'bad name' in the training literature. The main reason why they do is that they are upsetting. Errors provide feedback but feedback does not only have an informational side (what is not known yet?) but also a motivational component. Errors may, therefore, also demotivate the trainee (this is, of course, the important issue in Skinner's thinking). Depending upon their interpretation of the error feedback, the trainees may think that they are, for example, not smart enough to do it right. Errors—particularly mistakes but also slips with grave consequences (for example inadvertently deleting an important file)—are noticed and diagnosed on a conscious level. In such a case an attributional process sets in and the errors are interpreted according to who or what is at fault. Thus, negative attributions may develop; for example 'I am not able to learn to use such a complicated machine because I am too dumb' (Peterson and Seligman, 1984). However even when in a psychological sense benign attributions follow an error, one may lose fun and motivation in dealing with a system. A person may then 'leave the field' and stop working with this system.

In one of our studies (Frese *et al.*, in preparation), in which one part of the training consisted of error training, we presented subjects with heuristics that implied that it was positive to commit an error. Our qualitative observations were here that the negative attributions of errors persisted in the beginning and were slowly replaced by a much more active attitude towards making an error. Errors were then seen as chances to learn rather than as stressors.

Nevertheless, besides reducing motivation, errors may produce stress and anxiety. This has several implications for the learning process. First, stress can be conceptualized to be another—emotional—task the person has to cope with. This means, that several tasks have to be dealt with at once. Since the trainee is overloaded with information anyhow that has to be processed on a conscious level, the working memory becomes even more overloaded. Therefore, new errors (both slips and mistakes) can occur. The implication is that error management (which leads to acquiring a more positive attitude towards errors in the learning process) may actually lead to a reduction of errors.

People also have the tendency to revert back to overlearned responses when they are under stress (Semmer and Pfäfflin, 1978). This means that responses that are already regulated on a lower level will be preferred under stress conditions: since much of adult training is really *re*training, reintellectualization of formerly automatized responses and a conscious regulation of them is required. Since errors produce stress, this process of reintellectualization becomes disrupted and lower-level regulation is preferred. Again new errors (particularly slips) may result.

One important disruptive consequence of errors is related to system feedback. Does the feedback give information to the trainee and reduce its potentially demotivating function? Feedback is informative when the trainee learns to answer these four questions: (1) What specific error did I make? (help in feedback interpretation); (2) How did I get into this? (the knowledge base for regulation is improved); (3) What do I have to do so that I will not make this error again? (plans, goals and monitoring are implicated here); (4) What do I have to do to get out of the error state? (coping with an error situation—again referring to goals and plans).

The last point is actually the most important. Much of the aversiveness of errors in the human–computer field stems from the fact that errors (slips and mistakes) produce states that are difficult to leave. The novice will sometimes be forced to use a cold restart of the machine, with the effect that much or all of what was produced up to this point is lost. This problem became evident when observing the learning of WordStar. Incomprehensible error consequences often made the subjects very fearful of using the menus. This in turn reduced exploration and prevented a correction of inefficient methods.

Therefore, the training programme has to concentrate on this issue (but not only the training programme—also the error messages of the system and particularly the manual; cf. Wendel and Frese, 1987). Training has to ensure that one learns strategies to get out of the error state.

In summary, there are potentially negative effects of making errors—they may demotivate and increase stress and anxiety and lead to 'points of no return'. These effects will dampen exploratory strategies leading to anxious sticking to one particular method that may not be efficient and useful. To reduce these negative effects and to maximize the positive ones, it is necessary to explicitly integrate the problem of errors into the training process and to develop strategies of error management.

The notion of error management: integrating errors into the training process

When describing the notion of error management, one can again refer to the steps in the action process and to the knowledge base for regulation in Figure 1.

Knowledge base for regulation

In order to undertake effective error management, the person has to know potential error sources (of the program) and error tendencies (of him- or herself). Thus, knowledge of error 'spots' of the software system and a certain amount of realistic self-perception of one's error tendencies are important. When these exist, one can develop better hypotheses once the error has been committed and better ideas of when and where to be careful because error-free performance is important.

Goal development and goal decision

Whether or not an error is seen as a challenge or as a stressor depends on user's goals. Similarly, attributions are different depending upon the goals. If the goal in training is not to be error-free, but actually to get to know potential problem areas of the program and one's own proclivities for errors, in other words if committing errors is seen as part of what one should do in the training process, then committing an error is not reducing motivation to work with the system but rather enhancing it. For something to be frustrating, it has to be seen as negative—as something to be avoided. If the production of errors is seen as something positive (note: in the training process) then errors are not frustrating any more. Furthermore, if the goal consists of making errors, there will be no negative attributions in the Peterson and Seligman (1984) sense, which only appear after negative events. When it is the goal to make errors in the training process, the necessary knowledge can be developed to reduce the amount of errors in actual work situations, to get out of error situations and to be less anxious when trying out new things and committing errors again.

Plan development and decisions

Two aspects are important here: developing strategies to learn from errors and strategies to get out of error states. Strategies to learn from errors include, for example, learning how to interpret error messages, to look things up in the manual and to actively recapitulate what one has learned from an error.

One problem for novices is often that one error is followed by the next one and that they get deeper and deeper into an error state. Therefore, strategies to get out of the error states have to be learned. These may include, for example, ways of jumping from one menu to the other, of being able to use the escape and UNDO keys or of knowing how to do a warm restart. At a more metacognitive level (level of abstract thinking) it implies that one learns not to give up too easily and to try various and diverse ways of dealing with error states.

Perception and interpretation of feedback

Error diagnosis is often difficult (or made difficult by a particular software). It is not always obvious what the error was, what the system feedback means and where the error stems from. Therefore, the trainees have to be taught how to diagnose their errors. This implies two aspects. The first is to learn how to construct one's own feedback since not all errors produce obvious error messages. For example, SPSS version 9 has led to many errors in the treatment of missing values, because the COMPUTE command needed an additional ASSIGN MISSING command to treat missing data correctly. However, there was no error message when the ASSIGN MISSING command was missing. Therefore, error diagnosis implies here that the trainee had to learn to look for these mistakes by watching the number of cases in the output closely. Thus, an explicit strategy of error diagnosis has to be taught.

Second, where does the error stem from? This is also not always obvious. Sometimes it was not the last command that really led to the error but one occurring a few commands before (for example when using the overwrite mode and then trying to use the backward delete in the MS-WORD word-processing system). The problem here is that people tend to cling to their first hypothesis even if many reasons speak against this. This has been variously described as the problem of mental set (Luchins and Luchins, 1959, Levine, 1971) or as cognitive hysteresis (Norman, 1984). Therefore, trainees have to be taught to develop a set of heterogeneous hypotheses about error causes, to explore these alternatives and to be more skeptical about their first set of hypotheses.

To include these aspects of error management in the training process implies the following:

1. In general, the training should teach and support an active and exploratory approach. This implies, for example, that the trainees should be encouraged to develop their own hypotheses and their own mental model of the system, to use 'risky' strategies, to look into aspects of the system not yet known, to experiment when they are not quite sure, etc. (cf. Carroll and Mack, 1985; Frese *et al.*, in press). Only within such a training strategy does it make sense to develop error management strategies. When the training is structured according to programmed learning principles, when the trainee has to follow the instructions passively or when only one 'correct' answer is allowed, error management strategies cannot be taught.
2. An error training should be integrated into training programmes (cf., for example, Greif, 1986). In each case this implies that the trainee has to make some kind of mistake and then learn to get out of the error state. In our own trainings and experiments (for example Frese *et al.*, in preparation) we have variously tried the following strategies: (a) The trainees were asked to make as many different mistakes as they could think of. (b) The trainees had to follow through mistakes that were often made by other trainees. (c) One trainee had to learn how to get out of the errors of a second trainee. (d) Error states were reproduced on the screen or on

photographs and the trainees were asked to tell us how the error had come about and how to get out of the problem again. (e) Potential errors and strategies to get out of these resulting error states were explicitly described in a manual. (f) The trainees had to solve problems that were much too difficult; they were bound to make errors. When an error appeared, they were asked to try to get out of the error state.

In this type of error training, it is useful to ask two people to work together because then the experience is less frustrating. The error training has several functions. First, the trainee becomes familiar with the system. This enlarges the mental model and gives boundary lines for metaphors. Second, the trainee becomes accustomed to the use of exploratory strategies. Since errors and their negative functions are the main reason for giving up to explore, an error training may be a counter-strategy against this tendency. Third, errors are seen as more natural after such an error training and, therefore, produce less anxiety and stress. The trainee simply gets used to making mistakes and, therefore, the errors lose some of their threatening characteristics. Fourth, error training helps the trainee to learn strategies for getting out of difficult situations that resulted from the errors. Since it is one of the most disturbing aspects of the training process for beginners—that they do not know what to do any more—error training will counter despair and tendencies to leave the field (that is stop working with this programme). It is important, of course, that strategies to deal with errors have to be explicitly taught. At the moment we know rather little about which strategies have to be taught and which ones are effective. Furthermore, we do not know which strategies are typically used by good and not-so-good trainees and users. Research is needed here.

3. Most people (particularly adults) come into training with the concept that it is bad to commit errors. Therefore, every error is seen as a sign of weakness, of misunderstanding, of non-intelligence, etc. To counteract this strategy, it pays to present explicit heuristics (Skell, 1972). These heuristics should, for example, say: it is good to make mistakes; look at the screen to find out what happened when an error was made; you can learn from errors. With the help of these heuristics, the goal changes from not-wanting-to-make-any-error to let-me-see-what-I-can-learn-from-this-error.
4. Training should progress in a specific pattern over time, some things needing to be taught in the beginning and some at the end. We can, again, refer to Figure 1. In the beginning, the trainee has to work on the higher levels of regulation for each and every command. This implies that the trainee is overwhelmed with information and the limits of central processing capacity are often reached. Giving error training at this time only increases this information overload. Therefore, it would be useless to do error training at the beginning of the learning process. As a matter of fact, some rather surprising research results (Carroll and Carrithers, 1984; Carroll and Kay, 1985) pointing to the positive function of giving the trainee little feedback on errors may be due to this overload at the beginning of training. During such a training, the trainee has to concentrate only on the necessary keys, new commands, etc., and does not have to deal with self-produced problems. However, not giving any error training at all may lead to other problems, discussed above. Therefore, we suggest that error training should be done in the middle of the training. This reduces the overload for the novice in the beginning but gives the trainee a chance to use the experiences made in this section of the training later.

CONCLUSION

We still know very little about the 'typical' errors that trainees make. From informal observation, we know that most beginners use a typewriter analogy when learning word processing. This leads, for example, to difficulties in learning how to delete a blank space or a new line command. The type of error is probably related to prior knowledge of computers, to the metaphors used and to prior knowledge of the tasks that have to be done with the system. However, all in all, we know very little in this area.[1]

It is useful to analyse errors with the help of the theoretically derived taxonomy. However, it is not always easy to put empirical examples of errors into the cells of Figure 1 since it is hard to decide what causes an incorrect action. Errors have to be seen in relation to the goals and ideas of the person and these cannot be observed directly. For example, the MacWrite commands 'scroll down' and 'scroll up' were very often mixed up. There are two symbols: an arrow pointing upwards and an arrow pointing downwards. Normally such errors (mixing up commands) would be described in the cell 'level of flexible action patterns/plan decisions'. By chance one of our subjects explained the underlying cause of her error. Instead of clicking on the downwards pointing arrow to scroll up the text she clicked on the upwards pointing arrow. She described her reason for doing this: 'Oh, I must have been thinking of using a typewriter because if I wanted to see the text below I had to pull the sheet of paper up making an upward hand movement. This is why I always click the upwards pointing arrow.' This would actually mean that this error should have been classified as belonging to 'knowledge base for regulation' since it was caused by a 'wrong' mental model.

[1]Therefore, we are currently involved in a large empirical research project to answer what type of errors are typical of novices, intermediates and experts and how error treatment can be improved in software design, in software quality control and in training (Project FAUST, cf. the Acknowledgements).

Empirically it proves helpful to regard the errors with the help of the two dimensions in Figure 1. One could, for example, observe a trend over time to move from the intellectual and sensorimotor level to the level of flexible action patterns. This is so because new material is first learned on the intellectual level and with practice regulated one level further down. At the same time, old inadequate behaviour patterns that are still regulated on the sensorimotor level are reintellectualized and therefore moved upwards and then with practice moved again downwards in the hierarchy of the levels of regulation. A second trend over the period of the training can be seen in the move from goal decisions and feedback interpretation to plan decisions. Goals do not have to be developed and decided upon each time after one has become a more skilful user of a system and feedback interpretation is easier because there is a better mental model. However, plan decisions are becoming more complicated because the skilful user has many more alternatives at his or her disposal.

One can also differentiate between the two computer systems. For example, errors on the sensorimotor level have more negative consequences when using WordStar than when working with MacWrite because MacWrite is better in the support of error management. Interestingly, most errors using the MacWrite system could be categorized to be due to the knowledge base for regulation because some metaphors were unclear or inconsistent (e.g. the ruler). WordStar errors most often related to problems in goal decisions and feedback perception or interpretation because of the complexity and intransparency of the menu system and the abstract naming of the commands.

Thus, the taxonomy of slips and mistakes can serve to present a theoretically derived order to the confusing heterogeneity of errors. Additionally, the taxonomy has some practical relevance. If it can be shown that, for example, WordStar leads to feedback interpretation problems, one can treat the problem specifically. Either one can improve the system and present more obvious system reactions on the screen, or one can introduce a specific lesson during training with the goal of making the trainee monitor the screen carefully. It would also be valuable to the trainer to know the potential errors and their causes to be able to anticipate these and to take them into specific consideration during training (compare our remarks on error training).

In our view, errors may have positive and negative functions in the learning process. In a way, our argument is that both Skinner and the cognitive theorists are right, the Skinnerians with their view that errors lead to little motivation and the cognitive and action theorists with their emphasis on what can be learned from errors. In contrast to Skinner, however, we argue for the chance to make errors (although it might be better to reduce the chances in the very beginning of the training process and with very anxious trainees). Moreover, we think that error management is one important goal of training. Since the trainee has to be able to use newly developed skills in real-life tasks

and since the trainee is bound to make some errors, he or she has to learn how to expect these errors and how to cope with them effectively. The concept of error management helps to develop the prerequisites for dealing with errors and to develop ways of incorporating the teaching of error management strategies into the training process.

ACKNOWLEDGEMENTS

This article was produced as part of a research project financed by a grant to the first author from the Humanization of Work Fund of the Ministry of Science and Technology of the Federal Republic of Germany (01 HK 806 7): Project FAUST (Fehler Analyse zur Untersuchung von Software und Training). Other researchers in this project were: F. Brodbeck, H. Peters, J. Prümper and D. Zapf. We would like to thank L. Kershaw and D. Zapf for critical comments on a draft of this article.

REFERENCES

Altmann, A. (1987) Direkte Manipulation: Empirische Befunde zum Einfluß der Benutzeroberfläche auf die Erlernbarkeit von Textsystemen. *Zeitschrift für Arbeits- und Organisations-Psychologie*, **31**, 108–14.

Bruner, J. S. (1960) *The Process of Education*. Cambridge, Mass.: Harvard University Press.

Carroll, J. M., and Carrithers, C. (1984) Training wheels in a user interface. *Communications of the ACM*, **27**, 800–6.

Carroll, J. M., and Kay, D. S. (1985) Prompting, feedback and error correction in the design of a scenario machine. *Proceedings of the CHI'85 Conference on Human Factors in Computing Systems*, San Francisco, pp. 149–54.

Carroll, J. M., and Mack, R. L. (1985) Metaphor, computing systems, and active learning. *International Journal of Man–Machine Studies*, **22**, 39–57.

Carroll, J. M., and Thomas, J. C. (1982) Metaphor and the cognitive representation of computing systems. *IEEE Transactions on Systems, Man, and Cybernetics*, **12**, 107–16.

Frese, M., and Peters, H. (1988) Fehlerbehandlung in der Software-Ergonomie: Theoretische und praktische Überlegungen. *Zeitschrift für Arbeitswissenschaft*, **42**, 9–17.

Frese, M., and Sabini, J. (eds.) (1985) *Goal Directed Behavior: The Concept of Action in Psychology*. Hillsdale, N.J.: Lawrence Erlbaum.

Frese, M., Stewart, J., and Hannover, B. (1987) Goal-orientation and planfulness: action styles as personality concepts. *Journal of Personality and Social Psychology*, **52**, 1182–94.

Frese, M., Albrecht, K., Altmann, A., Lang, J., Papstein, P. V., Peyerl, R., Prümper, J., Schulte-Göcking, H., Wankmüller, I., and Wendel, R. (in press) The effects of an active development of the mental model in the training process: experimental results on a word processing system. *Behaviour and Information Technology*.

Frese, M., Brodbeck, F., Heinbokel, T., Mooser, C., Schleiffenbaum, E., and

Thoeman, P. (in preparation) Errors in training computer skills: Experimental results on the positive function of errors.

Greif, S. (1986) Job design and computer training. *Bulletin of the British Psychological Society*, **39**, 166–9.

Hacker, W. (1973) *Allgemeine Arbeits- und Ingenieurpsychologie*. Berlin: VEB Deutscher Verlag der Wissenschaften.

Levine, M. (1971) Hypothesis theory and nonlearning despite ideal S-R-reinforcement contingencies. *Psychological Review*, **78**, 130–40.

Luchins, A. S., and Luchins, E. H. (1959) *Rigidity of Behavior*. Eugene: University of Oregon.

Norman, D. A. (1981) Categorization of action slips. *Psychological Review*, **88**, 1–15.

Norman, D. A. (1984) *Working Papers on Errors and Error Detection*. San Diego, Calif.: University of California.

Norman, D. A. (1986) Cognitive engineering. In D. A. Norman and S. W. Draper (eds.), *User Centered System Design*. Hillsdale, N.J.: Lawrence Erlbaum.

Peterson, C., and Seligman, M. E. P. (1984) Causal explanations as a risk factor for depression: theory and evidence. *Psychological Review*, **91**, 347–74.

Rasmussen, J. (1985) *Risk and Information Processing*. Denmark: Riso National Laboratory.

Schönpflug, W. (1985) Goal directed behavior as a source of stress: psychological origins and consequences of inefficiency. In M. Frese and J. Sabini (eds.), *Goal Directed Behavior: The Concept of Action in Psychology*. Hillsdale, N.J.: Lawrence Erlbaum, pp. 172–88.

Schulte-Göcking, H. (1987) Die Erfassung von unterschiedlichen Lernstilen. Unpublished thesis, Department of Psychology, University of Munich.

Semmer, N., and Frese, M. (1985) Action theory in clinical psychology. In M. Frese and J. Sabini (eds.), *Goal Directed Behavior: The Concept of Action in Psychology*. Hillsdale, N.J.: Lawrence Erlbaum.

Semmer, N., and Pfäfflin, M. (1978) *Interaktionstraining. Ein handlungs-theoretischer Ansatz zum Training sozialer Fertigkeiten*. Weinheim: Basel.

Skell, W. (1972) Analyse von Denkleistungen bei der Planung und praktischen Durchführung von Produktionsarbeiten in der Berufsausbildung. In W. Skell (ed.), *Psychologische Analysen von Denkleistungen in der Produktion*. Berlin: VEB Deutscher Verlag der Wissenschaften, pp. 13–100.

Skinner, B. F. (1953) *Science and Human Behavior*. New York: Free Press.

Skinner, B. F. (1968) *The Technology of Teaching*. New York: Meredith Corporation.

Volpert, W. (1974) *Handlungsstukturanalyse*. Köln: Pahl-Rugenstein.

Volpert, W. (1981) *Sensumotorisches Lernen: Zur Theorie des Trainings in Industrie und Sport*, Vol. 3. Frankfurt/M.: Fachbuchhandlung für Psychologie.

Wendel. R., and Frese, M. (1987) Developing exploratory strategies in training: the general approach and a specific example for manual use. in H.-J. Bullinger and B. Shacker (eds.), *Human–Computer Interaction—INTERACT '87*. North-Holland, Elsevier, pp. 943–8.

Wingert, B. (1985) *Workshop Mensch–Maschine-Kommunikation '85*. Arbeitsgruppe: Entstehung und Behandlung von Fehlern.