Errors in Working with Office Computers: A First Validation of a Taxonomy for Observed Errors in a Field Setting

Dieter Zapf
Felix C. Brodbeck
Michael Frese
University of Giessen

Helmut Peters
Technischer Überwachungsverein, Munich

Jochen Prümper
Jugendstr. 8, D-8000 Munich 80

An action-oriented taxonomy of errors in human–computer interaction in the office differentiated four classes: functionality problems, usability problems, interaction problems, and inefficient behavior. Functionality problems were differentiated in how they affect the action process. Usability problems were differentiated according to levels of action regulation and steps in the action process. For example, conscious strategies were differentiated from automatic ones. To examine the taxonomy's construct validation, several hypotheses regarding error-handling time, need for external support, complexity at work, and novices versus experts were tested in a field study of 198

This is part of the research project FAUST study (a German acronym of "error analysis for the investigation of software and training"). The project was supported by a grant from the Work & Technique Fund of the Ministry of Research and Technology of the Federal Republic of Germany to Michael Frese (01 HK 806 7) in collaboration with Technischer Überwachungsverein (TÜV), Munich. We would like to thank Paul Booth, Sharon Clarke, Siegfried Greif, Jim Reason, and two anonymous reviewers for their help and comments on earlier versions of this article.

Correspondence and requests for reprints should be sent to Dieter Zapf, Fachbereich 06 Psychologie, Justus-Liebig-Universität Giessen, Otto-Behaghel-Strasse 10, D-6300 Giessen, Germany.

clerical workers at 11 German companies and 7 small firms. A total of 1,749 errors were observed within a 2-hour period, 1,306 were rated concordantly by two re-raters. As expected, errors resulting from conscious regulation and functionality problems needed more error-handling time than errors resulting from more automatic actions. There were more thought and memory errors at workplaces with high complexity. The most external support was needed for knowledge errors. Novices committed more knowledge errors and experts more habit errors. Practical implications are discussed both for software development and training.

1. INTRODUCTION

Errors produce human and economic costs. They contribute to stress at work in that they can interrupt the train of thought, the action plan, and lead to negative emotions. The economic costs depend on the number of errors and the time expended in error diagnosis and recovery. Some experimenters have suggested that up to 50% of working time is spent on error recovery (Hanson, Kraut, & Farber, 1984; Kraut, Hanson, & Farber, 1983; Shneiderman, 1987). Smelcer (1989) suggested that one error alone—the "join" error in the program "Structured Query Language"—contributes to a loss of $53 million in the U.S. These results may be exaggerated but we know very little about the amount of time the average employee spends attending to errors in everyday work. In any case, even a percentage as small as 5% would contribute to an enormous loss of time. Thus, decreasing error-recovery time reduces both stress and economic costs: a result that combines economic criteria as well as humanization of work criteria.

Therefore, the topic of errors is of high importance in the area of human–computer interaction. Though errors are often used as performance measures, there have been few attempts to analyze the psychological aspects of errors in more detail or to relate errors to aspects of software design (e.g., Booth, 1990; Davis, 1983; Lewis & Norman, 1986; Norman, 1984; Rizzo, Bagnara, & Visciola, 1987).

Most of the evidence, so far, is based on data gathered in the laboratory or in case studies. Whereas experiments are concerned with aspects of internal validity and allow precise time keeping and accounting of errors, external validity is lower. If one wants to generalize to work life, this is particularly crucial. For example, experiments do not tell us enough about the extent of user errors, the time spent in error recovery, or how people use support in an error situation in the working world. Employees are usually familiar with and better equipped to work with a given software than experimental subjects. Furthermore, employees' tasks are usually more routine than laboratory tasks.

In order to understand user errors in the field, it would not be useful simply to keep a nontheoretical tally of all errors appearing at work. For example, a typing error clearly has a different status than an error due to lack of knowledge. Thus, it is necessary to develop a taxonomy of errors. The taxonomy that we developed is presented in this article. The following thoughts guided the development.

1. Our taxonomy shows several similarities with taxonomies developed by Rasmussen (1982) and Reason (1987a), but there are also some important differences, which are addressed in the discussion section.
2. A theoretical approach seemed necessary while at the same time being practically useful. Therefore, the taxonomy was based on action theory (Frese & Sabini, 1985; Hacker, 1986; Volpert, 1987).
3. Taxonomies depend very much on the purpose and the field they were developed for. In some cases, two or three categories might be enough. In other cases, it is helpful to have more differentiated subtypes. Because there are hardly any theoretical criteria on what and how many types of errors should be differentiated, practical criteria are important: A taxonomy proves its relevance if different error types lead to different practical consequences.
4. Many error taxonomies are theoretically very elaborated but were never investigated empirically (e.g., Arnold & Roe, 1987; Heckhausen & Beckmann, 1990). If someone tried to apply them in an empirical study, he or she usually had to collapse many error categories for the analysis of empirical material (e.g., Sellen's, 1990 application of Norman's, 1981 taxonomy of action slips). Thus, in order to get meaningful and reliable results, the taxonomy should not be too elaborated.
5. Even when error taxonomies were applied in empirical studies, there were hardly any attempts to investigate their reliability and validity. Therefore, we attempt to develop and validate a taxonomy based on action theory which can be reliably applied in the field of human–computer interaction. Results from a field observational study on errors in the use of office software in the workplace are presented. The practical implications of the taxonomy are discussed in the final section.

1.1 Definition of Errors

There are three elements to an action-theory-based definition of an error: (a) errors only appear in goal-oriented action; (b) an error implies nonattainment of a goal; and (c) an error should have been potentially avoidable (cf. Frese & Peters, 1988; Norman, 1984; Rasmussen, 1987c; Reason, 1987b, 1990).

(a) Without a goal there is no criterion to call an action erroneous. If a person presses keys at random, an error cannot be committed.
(b) Within an action-oriented approach, "error" implies the (temporal) nonattainment of a goal. This psychological definition is, for example, contrary to a phenomenological approach, which defines error as an "out-of-tolerance action." Because one could purposely deviate from externally im-

posed actions, this would not be considered "erroneous" from a psychological point of view (cf. Sellen, 1990).

(c) If an error is due to circumstances beyond a person's control, one cannot speak of an error (e.g., losing data because of sabotage or a blizzard). According to Norman (1984), unforeseen and unplanned events can be defined as accidents. The same applies if negative outcomes occur due to chance occurrences (Sellen, 1990). To speak of an error, the person should (at least to some extent) be in control of the action.

Errors are sometimes attributed either to the machine or to the user. Because an error depends on a goal, machines cannot commit errors because they do not have intentions of their own. Only when a machine is used to attain a certain goal, can errors occur. However, in some cases, for example, when the system crashes or when a software bug causes wrong results, then it seems, from the user's perspective, as if it is the computer which made the error.

On the other hand, errors are often attributed to (other) humans. They are considered to be the unreliable part of the man-machine system. However, this point of view cannot be justified theoretically, and as Lewis and Norman (1986) pointed out, it could actually be the computer, which says in a way, "I am sorry, I do not understand your command." Still, the negative connotation and assignment of blame is a real issue in the question on errors (cf. the fundamental attribution error of Nisbett & Ross, 1980).

In contrast, a scientific point of view would emphasize that an error is always due to a *mismatch* of the human to the computer system (Rassmussen, 1985, 1987a). It is a moot point to assign blame either to the machine or to the human, for example, when a file gets lost. If there is no back-up file in the system, this implies that the system should be improved. Nevertheless, it may also be useful to teach the person to be more careful when deleting files. In any case, the error was the result of the mismatch between the system and the person.

1.2 A Taxonomy of Errors

There are several forms of mismatch. It is useful to distinguish user, task, and program as components of the man–machine system (cf. Figure 2). Within this system, the mismatch of functionality and the mismatch of usability can be localized (Zapf, Brodbeck, & Prümper, 1989).

1.2.1 Functionality Problems. Functionality problems refer to the mismatch between task and program. This mismatch occurs if it is not possible to achieve one's goal as planned with a particular computer program. Functionality problems could be classified according to technical criteria. For example, Bons (1985) distinguished functionality problems (he called them system errors) according to their origin in the process of software development. Such a criterion cannot be observed after the release of the software, of course. Therefore, we decided to differentiate functionality problems according to their consequences for the action process:

(a) The worst case is that one has to give up or change one's goal. This is called *action blockade.* (E.g., a user tried to e-mail a formatted letter according to the description in the manual, however, this was not possible. Thus, the user gave up this goal and accepted the standard format.)
(b) In the case of *action repetition,* a part of one's work is lost and has to be redone. (E.g., the user does not get output as usual, and is forced to fill in several pages again.)
(c) In the case of *action interruption,* the user is interrupted by the system (e.g., a breakdown), but is able to continue his or her work, usually after some additional work.
(d) In many cases, users already know the weaknesses of their system and compensate for them by additional actions. This is called *action detour.* (E.g., in one system we observed that the hyphenating function caused many errors. Therefore, the users did not use the automatic mode, but used a manual mode that enabled them to reject erroneous hyphens immediately. However, this cost a considerable amount of time.)

1.2.2 Usability Problems. Even if the functionality of a computer program is sufficient for a certain task, errors still occur. They can be caused by a mismatch between user and computer, called *mismatch of usability.* From an action-theory perspective, mismatches of usability can be differentiated according to steps in the action process and different levels of action regulation (Figure 1 presents a simplified model using these two theoretical dimensions).

A common assumption in action theory is that actions are goal-oriented (Frese & Sabini, 1985; Hacker, 1986; Volpert, 1987). Within this approach, the action process comprises goal and plan development, the execution of actions as well as monitoring, and feedback processes (Frese & Stewart, 1984; Hacker, 1985; Norman, 1986). Goals and plans have to be developed first, then the plans are executed. To do this, subplans are usually stored in memory and recalled as needed to achieve subgoals. Monitoring environmental contingencies and one's own procedure is a prerequisite in order to use the correct subplan at the right time. Finally, feedback is given whether or not goals or subgoals were attained.

Similar to Hacker (1986) and Rasmussen (1982, 1985), three levels of action regulation are distinguished within the framework of hierarchically (or better, heterarchically) organized action plans and goals. The basic concept is the functional unit. It assumes that actions are regulated by their goals similar to a cybernetic control loop (cf. Miller, Galanter, & Pribram's, 1960 TOTE unit). The goal initiates a certain action, and feedback processes tell whether the goal was attained or not. Each goal can be divided into subgoals. Subgoals again can be divided into sub-subgoals and so forth. Thus, a pyramid of goals, subgoals, and sub-subgoals emerges. The functional units for each goal are hierarchically nested and control the sequence of actions. (This hierarchic-sequential model of action regulation is described in detail in Frese & Zapf, in press; Hacker, 1986; Volpert, 1982). Within the pyramid of goals and referring plans, three levels of action regulation are distinguished (Hacker, 1986). On the intellectual level (Rasmussen's, 1982 "knowledge based"), complex analyses of situations and problems are regulated. New plans are designed comprising the analysis of possible goals and environmental conditions, problem solving, and decision making. Regulation on this level is conscious and predominantly works in a serial mode, interpreting feedback step-by-step. At the level of flexible action patterns (Semmer & Frese, 1985; Rasmussen's 1982 "rule-based") actions are regulated by action patterns, which can be considered as schemata (cf. Norman, 1981). These are ready-made action programs available in memory. They have to be specified by situationally defined parameters and activated and integrated into an action chain for a specific situation. Processes at the level of flexible action patterns can be regulated consciously but they need not necessarily be conscious. The sensorimotor level (Rasmussen's 1982 "skill-based level") is the lowest level of regulation. Stereotyped and automatic movement sequences are organized without conscious attention (cf. Schmidt's, 1975 motor schemata). Regulation takes place with the help of proprioceptive and exterioceptive feedback. This type of regulation is largely unconscious. Conscious regulation cannot modify action programs at the sensorimotor level. At most, it can stop performance.

In addition to the levels of action regulation, there is the *knowledge base for regulation* (no equivalent in Rasmussen, 1982). There are at least three aspects of the knowledge base for regulation: knowledge of facts, knowledge of procedures, and understanding in the sense of mental models (cf. Anderson, 1983; Gentner & Stevens, 1983). This knowledge is used to develop goals and plans.

In Figure 1, the taxonomy of problems of usability is depicted (adapted from Zapf, Brodbeck, & Prümper, 1989). Descriptions of the different errors outlined in Figure 1 follow (for details, see Frese & Peters, 1988; Zapf et al., 1989). The terms used for each taxon are suggestive only, more important for the understanding of the taxa are the two dimensions (steps in the action process and levels of action regulation):

Knowledge errors occur when one is unable to do a task with the computer because one does not know certain commands, function keys, rules, and so forth.

On the intellectual level of action regulation, goal and plan development, as well as analyses of situations, are rather complex.

Thought errors occur when goals and plans are inadequately developed or when wrong decisions are made in the assignment of plans and subplans although the user knows all the necessary features of the system. (E.g., a user wants to design a table with 12 columns. He wants it to fit on one page. After some work he realizes that the columns are too wide. He has to change them all.)

Memory errors occur when a certain part of the plan is forgotten and not executed, although the goals and plans were initially correctly specified. (E.g., a person conceptualizes a spreadsheet with several rows and columns.

Knowledge base for regulation			
Knowledge errors			
Regulation level	Steps in the action process		
	Goals/ planning	Monitoring	Feedback
Intellectual level of regulation	Thought errors	Memory errors	Judgement errors
Level of flexible action patterns	Habit errors	Omission errors	Recognition errors
Sensorimotor level of regulation	Sensorimotor errors		

Figure 1. A taxonomy of usability problems (after Zapf, Brodbeck, & Prümper, 1989).

After printing it she realizes that she has forgotten a column that was originally planned.)

Judgement errors appear when one cannot understand or interpret the computer feedback after an input. (E.g., a person receives the message: "Not enough space." She does not know whether this message refers to the working memory or to the space on the hard disk.)

Errors on the level of flexible action patterns occur when well-known actions are performed.

Habit errors imply that a correct action is performed in a wrong situation. (E.g., a person switches over from the use of one word-processing system to the next one. Doing this, she still uses the function keys that were correct in the former system but are now incorrect.)

Omission errors happen when a person does not execute a well-known subplan. This is most likely when the person is interrupted in an action plan. (E.g., a person forgets to save a file although he or she usually does this routinely at the end of a session.)

Recognition errors appear when a well-known message is not noticed or is confused with another one.

Sensorimotor errors are placed at the sensorimotor level. There is only one category here because, at this level, it is empirically difficult to differentiate among planning, monitoring, and feedback. (E.g., touching the wrong function key or wrong mouse movements are categorized here.) Most of the typing errors would be categorized as sensorimotor errors. However, for our purposes, only those errors that did not occur in the context of writing continuous text were placed in this category.

1.2.3 Inefficiency Errors and inefficient behavior have a large conceptual overlap. A detour to reach a goal may be conceptualized as an inefficiency but also as an error, because usually one's goal is to proceed in the most straightforward manner. If there is any differentiation between the two terms, inefficiency could be formally defined as any deviation from an optimal action path (cf. Oesterreich, 1981). The optimal action path is the shortest way from a given situation to the goal. But even this conceptualization has its difficulties. It may be more cost effective, in the psychological sense, to use a strategy that is inefficient in the formal sense because setting up plans and differentiated calculations for them imply psychological costs as well (Schönpflug, 1985).

Pragmatically, one's behavior is described as inefficient when one is successful in attaining a goal that should have been more easily attained in a more "direct" manner (direct implies that each subaction leads closer to the goal and no subaction has to be undone).

Two groups of inefficiencies were differentiated: Inefficiency due to lack of knowledge, which implies that an inefficient path is followed because one does not know a better way, and inefficiency due to habit. The latter implies that one uses routines although the person knows that there would be a more efficient path available.[1]

1.2.4 Interaction Problems They imply that the errors occur not because of individual problems with the computer, but because there is a mismatch between individuals. Although the individuals' actions are more or less correct, an error occurs because of an organizational lack of coordination, unclear task allocation, or because of lack of communication between individuals (e.g., a user deletes a file that was still needed by another person). The error categories used in our study are summarized in Figure 2.

One could question whether the term "error" is really suitable to embrace all the phenomena discussed in this article. Can it be called an error if a person does not know whether to press A or B, and, therefore, does nothing? The person obviously has a problem. However, if he or she does not know which button to press and presses A when B is correct, it clearly is a knowledge error. Psychologically, there is not much difference. Actually, there was a lot of discussion about the use of the term "error" in the research project. Other candidates would have been "mismatch" or "problem." Though mismatch seemed to be most preferable, it also

---

[1]It should be mentioned that we considered the level of qualification in our empirical investigations. When a menu-driven word processor is used, it is not considered inefficient to work with the menu instead of the quicker function keys and commands.

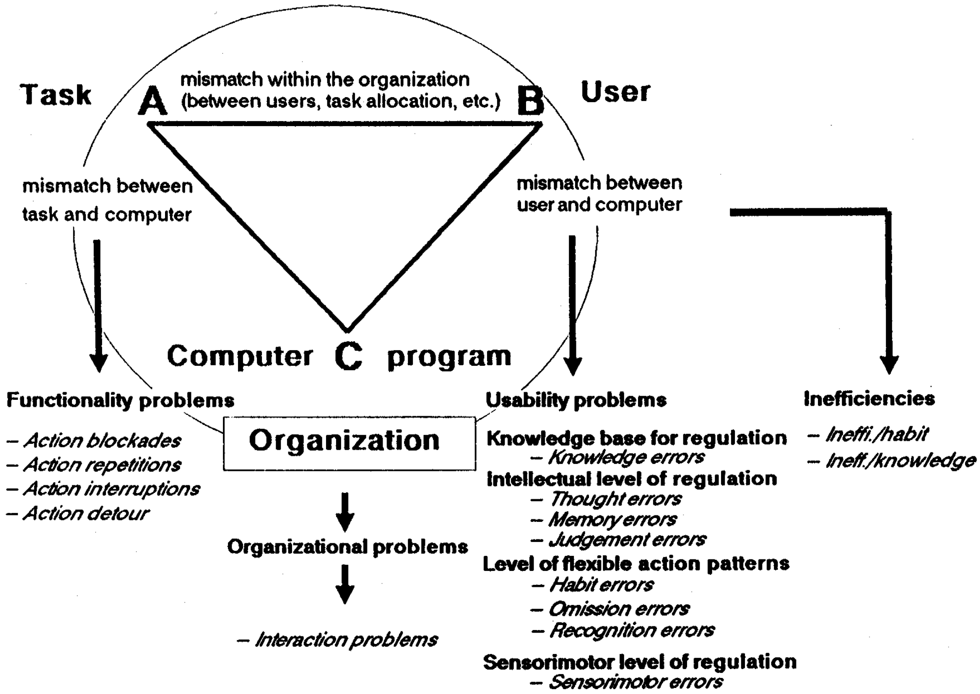

Figure 2. Overview of error categories (italicized) used in this study.

seemed to be too vague, particularly in communication with practitioners. This is partly a translation problem: There is no really suitable equivalent for mismatch in German. Knowledge errors and thought errors could be called knowledge or thought problems as well. They meet common definitions of a problem (e.g., that there is a barrier between current state and goal attainment; cf. Dörner, 1976). However, this would not be true for errors at the lower levels of regulation. For these reasons, we decided to keep the term "error."

2. VALIDITY ASPECTS OF THE ERROR TAXONOMY

Evaluating a taxonomy can be done analogously to evaluating an empirical scale (Fleishman & Quaintance, 1984). Thus, the validity of our taxonomy will be discussed in terms of construct validity (Cronbach & Meehl, 1955). There are four groups of hypotheses that we want to use for construct validation.

2.1 Error-Handling Time

Before an error can be recovered, it needs to be detected. Not all errors are detected (Bagnara, Stablum, Rizzo, Fontana, & Ruo, 1987; Rizzo et al., 1987). Not all detected errors can be explained by the users. Sometimes an error can be corrected without knowing how it came about. After detection, the following processes take place to cope with an error (Reason, 1990; Zapf, Lang, & Wittmann, 1991): (1) error diagnosis with its components of error detection and error explanation; and (2) error recovery with its components of planning of recovery and recovery action.

We will not discuss this outline in detail because it was done in Brodbeck, Zapf, Prümper, and Frese (1990) and Zapf, Lang et al. (1991). For our purposes here, it is important to note that all of these error-handling processes take time. Error-handling time is the time interval starting after error detection and continuing to the point of successful correction or giving up the attempt to correct. We did not include the time between error occurrence and error detection. First, in many respects, an error is only of psychological relevance to the user once it has been detected. Second, in many cases, the time point of error occurrence can only be reconstructed after it has been detected. Because we had no keystroke protocols in the field study, this could not be done reliably.

The intellectual level of regulation implies conscious thoughts and more complicated errors. Therefore, handling time should be longer here than on the lower levels of regulation. Furthermore, knowledge errors imply a need for additional time because one either has to look up information (e.g., in a handbook) or explore to find the correct procedure. Finally, the correction of functionality problems should be time-costly at least as long as one does not already know how to compensate for them.

2.2 Support and Error Handling

There are two types of resources to draw on when correcting an error: internal (one may explore solutions systematically or unsystematically) or external (one may consult a handbook, coworker, supervisor, advisory center, or the help menu; more on this in Brodbeck et al., 1990).

The lower levels of regulation were defined to pertain to routine actions. Therefore, the person committing an error should be able to correct it without external support because the person knows the correct action and has performed it frequently. (This is not always so; an error at the lower levels might also get one into an unknown state, but this is not the typical situation). In contrast, errors on the intellectual level may call for more support. Most external support should be needed for the handling of knowledge errors, because, by definition, lack of knowledge is implicated here and help from external sources is, therefore, often necessary. (This is not always the case because sometimes there are only two alternatives; after one alternative has been tried, one knows which one is correct.)

2.3 Errors and Complexity of Work

Complexity in this article is regarded as the amount of regulation requirements (Frese, 1987a; Oesterreich & Volpert, 1986; Volpert, 1987). This means that at workplaces with low complexity, action regulation is constrained to the lower levels of regulation (sensorimotor level and level of flexible action patterns). In this case the tasks are routine to the user. Goals are familiar and only simple decisions have to be met. For example, writing simple letters with a word processor only requires regulation at lower levels. High-complexity jobs include regulation at the intellectual level. Unfamiliar goals and plans have to be developed and decisions at this level are difficult. Usually, high-complexity jobs comprise routine task aspects as well, and thus, also require regulation at the lower levels. According to this concept of complexity, it is obvious that error types depend on the complexity of work. We hypothesize that there are fewer errors on the intellectual level if the task has little complexity. Reason (1987b) has argued that situations of both high and low complexity show a similar number of slips (in our terminology, errors on the lower levels of regulation). This is in line with the assumption in action theory that complex tasks usually comprise subgoals and plans regulated at lower levels. Therefore, it was hypothesized that complexity should not affect errors on the lower levels of regulation (flexible action patterns and sensorimotor skills).

Additionally, one might expect that functionality problems will occur more often when the job is complex, and therefore, in most cases, complex software systems have been utilized. These complex software systems should be prone to present more functionality problems.

2.4 Errors: Novices Versus Experts

We assume that novices commit more errors on the intellectual level of action regulation. The clearest differences, however, should occur for knowledge errors: Novices do not have enough knowledge about functions, commands, and so on (Frese & Peters, 1988). Zapf and Frese (1989) argued that thought errors are related to the work task as a whole, whereas knowledge errors are more specifically related to the interaction aspect with the computer. The subjects in our study were all more or less experts on their work tasks. However, some of them were novices with regard to the computer program they were using. Thus, there should be more clear-cut differences between novices and experts in knowledge errors than in thought errors. In contrast, experts might make more errors than novices on the lower levels of regulation because many of their actions are done routinely (Adelson, 1984; Frese & Peters, 1988; for details, see Prümper, Zapf, Brodbeck, & Frese, in press).

3. METHODS

3.1 Subjects

The taxonomy of usability and functionality problems, as well as interaction problems and inefficiencies, were used in a study of 259 users of office software from 15 departments in 11 different public and private companies and 7 small firms in the southern part of the Federal Republic of Germany. In 9 of the 15 departments we tried to get the full population. Actually, on average, 85% of the persons in these departments participated in our study. Mainly, there were three reasons why people did not take part: First, because of workload, and second, because they seldom worked with the computer and/or the computer work could not be organized during the time we spent at the respective company. Additionally, some of these persons argued that they had had too little computer experience. A third reason was that people were ill or on vacation. In all, very few persons refused to participate actively.

In one company, 29 of 260 persons were selected by an internal trainer according to the criteria job level and computer expertise (the time worked with the particular computer system). Two persons refused and were replaced by other ones. In the remaining four departments, an unsystematic selection of participants took place and it is difficult to evaluate the rate of refusals. A brokerage association manager selected all the small brokerage firms where he knew computer equipment was used. Thirty of them received a letter of invitation to participate. Seven of them (23%) took part in our investigation.

The mean age of the sample was 31.1 years; 73% were women, 27% were men. The work tasks included typewriting, secretarial work with office communication systems for several administrative purposes, specialized tasks in insurance and public administration with integrated systems including large data bases and word processing, and tasks from lower management with, for example, spreadsheet programs to administrate the employment and absenteeism of workers. There were no workplaces where the use of programming languages like Cobol, PL/1, and so on, was required. Sixty-nine percent of the people had worked more than 1 year with computer systems; 13% had worked fewer than 6 months with computers and could therefore be considered novices.

For organizational reasons, not all of the 259 subjects could be observed. There are observational data from 198 persons, questionnaire data from 232, and both sets of data from 174 persons.

3.2 Measures and Procedures

The subjects were observed, interviewed, and received a standardized questionnaire asking for work characteristics, reactions to errors, help requested for error management, and level of computer expertise. For this study, questionnaire data were used to differentiate different levels of computer expertise.

Our research is one of the first field studies on observing errors in human-computer interaction in the workplace. In this type of research, there are higher constraints than in the laboratory. For example, we were not able to use keystroke protocols, videos or stopwatches of the subjects' work. Because we observed 16 different programs with different computer systems, keystroke protocols would have been difficult to get merely because of technical problems. Additionally, German firms and research projects have a strict policy that the shop stewards must give their consent to the research. Any research project that could have been used to advance keystroke protocols as performance indicators or stopwatches would not have been permitted by the shop stewards. The same reasoning applied to videos as well. For this reason, great care was taken to develop a useful observational procedure with approximate indicators that could be observed.

The observers participated in a 3-day training course that included a discussion of theory and procedures, as well as an evaluation of a set of errors that had been previously collected from subject matter experts (Brodbeck, Prümper, & Zapf, 1988). Furthermore, the trainees practiced for 1.5 days and observed an experienced observer before working with the subjects. Fifteen observers took part in the study: 5 members of the research team, 6 psychology students, 2 economics students, and 2 industrial psychologists.

Before observing the subjects, a short job analysis was done with the VERA instrument Part A of Volpert and colleagues (Oesterreich & Volpert, 1986; Volpert, Oesterreich, Gablenz-Kolakovic, Krogoll, & Resch, 1983) to describe the tasks that were performed at the observed work place. This was also necessary in order for the observers to receive an overview of the tasks they were to observe. Because errors are always defined by the goal, the knowledge of the task structure helped the observers to determine what goals the subjects were trying to achieve.

In a second step, the subjects were observed doing their normal work with the computer. In some cases, people also did many noncomputer tasks, and it was necessary to ask them to do their computer tasks during the observation period. The observation period lasted for 2 hours. The observer sat next to, or behind, the subject in order to see both the screen and the keyboard. The observers had two different protocol sheets. One protocol kept a record of the computer tasks that the subjects worked on and the amount of time required. A second protocol was kept for all errors and subsequent recovery actions. Here, the observers gave a written description of each error.

Because of economic and organizational reasons, it was not possible to carry out observations by two observers in each case. Therefore, to assess the reliability of the observational measures, a small study with two observers was carried out (Herbrich, Frese, & Prümper, 1991). Twenty-three employees were observed by two researchers simultaneously with the same procedure as in the main study; 152 errors were recorded by both researchers, 100 errors were rated concordantly. This is an agreement of 65.8% or a Cohen's (1960) kappa of .62.

In the field study reported here, 1,749 errors were recorded by the observers. Based on their verbal descriptions, two members of the research team independently assigned the descriptions to the error categories described before. Interrater agreement for these two raters was 74.7%. That is, 1,306 errors were re-rated concordantly by two members of the research team. Because random agreement occurs seldomly with 15 categories, the total agreement is an appropriate statistic (Bortz, 1984). A concordance rate of 74.7% does not seem to be very high; however, even experimental studies (Bagnara et al., 1987; Rizzo et al., 1987) have shown comparable levels of agreement using videos and keystroke protocols. This procedure of re-rating the written protocols is equivalent to using two observers at the work site (Prümper, 1991). Unless otherwise mentioned, only those errors that showed agreement were analyzed. Thus, there were 1,306 errors or subsamples analyzed in this study. Prümper (in press), who analyzed the interrater reliability in more detail, identified two major reasons for nonagreement. First, some of the descriptions of error observations gave too little information about the software system. This led to confusions within the different categories of functionality problems, and between functionality problems and knowledge errors. Second, it was sometimes unclear how routine the task was to the user. Therefore, the re-rater sometimes chose the same steps in the action process, but different levels of regulation (e.g., they mixed up omission errors and memory errors).

The observers also recorded how the subjects handled each error, whether they used external resources like manuals, and the amount of time needed for error handling (Brodbeck et al., 1990). In 96% of all cases, the use of external resources (manuals including self-written short papers, help systems, colleagues, or advisory service) could be rated in agreement. This is equivalent to Cohen's (1960) kappa coefficient of .81 (Herbrich et al., 1991). It was not possible to use a stopwatch (fear of being timed for performance would impede the cooperation of the subjects and the shop stewards). Therefore, error-handling time was recorded as a rough estimate by the observer. Because these are ordinal data, it was necessary to analyze the results on handling time with chi square. This was routinely done in our analyses. However, the tables are difficult to interpret. For expository purposes, we will, therefore, present the data differently. The data were transformed into a real-time scale. This was done with the rough formula in Table 1.

The results can then be presented as means which render them more interpretable.[2] The 10 min or longer category leads to a conservative estimate of the total handling time because many subjects needed a much longer handling time, in some cases more than 30 min. Error correction time was rated concordantly in 80% of all cases by two observers (Herbrich et al., 1991) with a Spearman's correlation coefficient of .68.[3]

Additionally, the observers rated the complexity of the job. A four-item scale of complexity was computed. The items had a range from 1 to 5 (Cronbach's $\alpha = .81$; for details see Zapf, 1991). This scale has shown high validity in former studies

Table 1. Transformation of Error-Handling Time

Observer Estimates	Immediate	<2 min	<5 min	<10 min	≥ 10 min
Transformation	.25	1	4	8	12

[2] The chi-square tables are available from the first author upon request.

[3] In a laboratory study where we applied both observations and keystroke protocols, the obervers' estimation of correction time correlated ($r = .75$) with the data collected by keystroke protocols (Lang, Prümper, Wittmann, & Zapf, 1991).

(Semmer, 1984; Zapf, 1989). The sample was trichotomized at the level of workplaces (complexity = < 2.75, 2.75–3.25, or > 3.25).

4. RESULTS AND DISCUSSION

This discussion will first present overall results for the taxonomy. Second, the hypotheses regarding construct validity will be tested.

4.1 Distribution of Errors

The distribution of errors in the taxonomy is presented in Table 2. There were many inefficiencies (in all, 16% of the problems were classified in the two subclasses of inefficiency).

As for usability problems, sensorimotor errors were the most frequent ones (16.6%) as one would expect (*note:* typing errors in text writing were excluded). There were also many thought errors (10.3%) and knowledge errors (9.5%). Judgement errors were relatively infrequent. Apparently, errors related to not understanding the system's reactions appeared only rarely.

Table 2. Frequencies of Error Types

	Frequency	%
Inefficiency		
Inefficiency/habit	93	7.1
Inefficiency/knowledge	116	8.9
Usability problems		
Knowledge base for regulation		
Knowledge errors	124	9.5
Intellectual level of regulation		
Thought errors	135	10.3
Memory errors	61	4.7
Judgement errors	26	2.0
Level of flexible action patterns		
Habit errors	115	8.8
Omission errors	123	9.4
Recognition errors	48	3.7
Sensorimotor level of regulation		
Sensorimotor errors	217	16.6
Functionality problems		
Action blockades	19	1.5
Action repetitions	28	2.1
Action interruptions	30	2.3
Action detours	143	10.9
Interaction problems	28	2.1
Total	1,306	100.0

Functionality problems comprised 16.8%, of which action detours were relatively frequent (10.9%). This means that the subjects knew the functionality problems beforehand and compensated for them. This is in line with the fact that most subjects had a good knowledge of their system. There were few action blockades. This is understandable because organizations do not tolerate technical systems that do not allow the performance of work tasks.

Only few interaction errors appeared. This may be due to the fact that networks were not used heavily in our sample.

One problem discussed in Peters, Frese, and Zapf, 1990 is whether the distribution of errors depends on computer systems and interfaces. In fact, there are, for example, considerable differences in the frequency of functionality problems (ranging from 0–4.9 functionality problems per observation period for the 16 observed systems). There were also different error distributions for different dialog forms, suggesting that the taxonomy can be used for the evaluation of computer systems and interfaces. However, no indication could be found that these differences did affect the following validity analyses.

4.2 Testing the Construct Validity Hypotheses

4.2.1 Recovery Time. Total error-handling time across all subjects amounted to 8.5 min of the observation time[4] (minimum = .25 min, maximum = 85.25 min; SD = 12.36 min). As mentioned in the method section, employees performed some noncomputer work during the observation period as well. During the observational period the average computer work time was 85 min. This means that our subjects spent 10% of their observed computer work time with error handling. Given that most subjects were quite experienced (the mode of experience with the systems was 1–2 years) and the fact that even the novices in our sample had enough skills to carry out their tasks, this is a high percentage. Moreover, it is a conservative estimate (see the method section).

The distribution of error-handling time across the taxonomy is presented in Table 3. Error categories that required no error handling were excluded from this Table (inefficiency and action detours). The results were in line with our hypotheses. Lower levels of regulation errors were corrected very quickly. Sensorimotor errors needed the least amount of time. None of the errors on this level of regulation needed more than 5 min. Similar results appeared for errors at the level of flexible action patterns. On average, these errors were corrected within 1 min or less. In contrast, error handling on the intellectual level of regulation required more time. Similarly, knowledge errors were difficult to handle. Most time was spent on judgement errors, but note the high standard deviation suggesting a high variability.

[4]This analysis was based on all errors that were corrected, because we wanted to know the overall correction time independent of the kind of error. Therefore, interrater concordance did not play any role for this computation.

Table 3. Error Types, Means, and Standard Deviations of Error-Handling Time

Error type	No.	M (min)	SD
Usability problems			
Knowledge base for regulation			
Knowledge errors	111	2.38	3.42
Intellectual level of regulation			
Thought errors	123	1.44	1.81
Memory errors	59	1.50	2.02
Judgement errors	23	4.12	4.00
Level of flexible action patterns			
Habit errors	110	.50	.58
Omission errors	123	.60	.71
Recognition errors	44	1.01	1.94
Sensorimotor level of regulation			
Sensorimotor errors	213	.46	.70
Functionality problems			
Action blockades	12	5.33	4.50
Action repetitions	27	2.15	3.14
Action interruptions	30	1.58	2.28
Interaction problems	25	1.71	2.58
Total	900	1.23	2.17

Note. $F\ (11, 900) = 19.26, p \leq .01$.

Thus, our hypothesis that handling time is highest for errors on the knowledge base for regulation and on the intellectual level of regulation was borne out by the data. As predicted, functionality problems also took a long time to correct with action blockades having the longest error-handling time.

4.2.2 External Support and Error Handling. Table 4 shows a comparison for errors corrected with and without external support. External support included the use of a help system, a manual, a coworker, or the advisory center. In Table 4, all errors that were not handled by the users alone were collapsed. Error categories that needed no correction (inefficiency and action detours) and those that could not be handled (3.6%), were excluded in this Table (cf. Brodbeck et al., 1990).

In all, 12.7% of the errors were corrected with the help of external support. Because the subjects knew the systems quite well, the amount of external help needed seems rather high and underscores reports of how important external supports are in human–computer interaction (Bannon, 1986; O'Malley, 1986; Scharer, 1983).

The distribution across the taxonomy supports our hypothesis. Only about 44% of all knowledge errors could be corrected without external support. A similar picture was true for judgement errors: More than 40% could only be corrected with external support. This is understandable because judgement errors are similar to knowledge errors. They depend on the knowledge of the computer system: A person who knows the system quite well will have fewer difficulties interpreting and

Table 4. Error Types and Internal Versus External Support for Error Handling (Frequency and Row Percentage)

	Support			
	Internal		External	
	Frequency	%	Frequency	%
Usability problems				
Knowledge base for regulation				
Knowledge errors	49	44.1	62	55.9
Intellectual level of regulation				
Thought errors	110	89.4	13	10.6
Memory errors	52	88.1	7	11.9
Judgement errors	13	56.5	10	43.5
Level of flexible action patterns				
Habit errors	106	96.4	4	3.6
Omission errors	119	96.7	4	3.3
Recognition errors	43	97.7	1	2.3
Sensorimotor level of regulation				
Sensorimotor errors	213	100.0	0	0.0
Functionality problems				
Action blockades	7	58.3	5	41.7
Action repetitions	25	92.6	2	7.4
Action interruptions	29	96.7	1	3.3
Interaction problems	20	80.0	5	20.0
Total	786	87.3	114	12.7

Note. χ^2 $(11, N = 900) = 273.96, p = .000.$

understanding system reactions. As predicted, thought errors and memory errors required more support than errors on lower levels of regulation. In contrast, only very few of the errors on the level of flexible action patterns required external support. All sensorimotor errors could be handled without external support.

Functionality problems present a differentiated picture. Action repetition and action interruption could nearly always be handled without support. However, this was not true for action blockades. Most of those could not be corrected at all. Only 58% could be handled without external support.

Twenty percent of the interaction problems required help. In those cases, it was usually necessary to consult with coworkers.

4.2.3 Errors and Complexity of Work. As described in the method section, the sample was trichotomized (low, medium, and high work complexity). In Table 5, errors are displayed at workplaces with low (n = 63) and high complexity (n = 60). In this computation, the number of errors for 1 computer work hour were compared (because there were 15 error categories, the mean error frequencies per hour were, of course, rather low). Our hypotheses was supported: There were more thought errors and memory errors for subjects who worked in more complex jobs. There were no significant differences for knowledge errors. This supported our hy-

Table 5. Complexity at Work and Error Types

	Mean No. of Errors/Hour Complexity		
Error Type	**low (n = 63)**	**high (n = 60)**	t
Inefficiencies			
Inefficiency/habit	.26	.42	−.82
Inefficiency/knowledge	.47	.30	.82
Usability problems			
Knowledge base for regulation			
Knowledge errors	.36	.48	−.84
Intellectual level of regulation			
Thought errors	.31	.75	−2.63**
Memory errors	.20	.35	−1.80*
Judgement error	.09	.04	1.11
Level of flexible action patterns			
Habit errors	.52	.32	1.51
Omission errors	.57	.42	1.13
Recognition errors	.23	.16	.91
Sensorimotor level of regulation			
Sensorimotor error	1.08	.76	1.38
Functionality problems			
Action blockades	.06	.08	−.33
Action repetitions	.08	.13	−.81
Action interruptions	.05	.11	−1.02
Action detours	.87	.42	1.01
Interaction problems	.20	.10	.96

*$p < .05$, one-tailed. **$p < .01$, one-tailed.

pothesis that thought and memory errors depend more on the work task as a whole. Knowledge errors, on the other hand, are probably more related to the specific computer system and do not depend on knowledge problems about the task. The errors on the lower levels of regulation showed no differences in number of errors for different complexity levels. Thus, work complexity did not affect lower regulation-level errors. No differences were found for functionality and interaction problems.

The theoretical importance of this result is that complexity of work as an organizational variable affects error types differentially. Thus, organizational aspects can have an influence on the specifics of how a person deals with a particular system. Therefore, software design has to be integrated into organizational and work design issues (cf. Frese, 1987b; Hacker, 1987; Ulich, 1989, 1991).

4.2.4 Errors by Novices and Experts. The next question was whether there were differences between expert users and novices. To examine this, two extreme subsamples were differentiated: One included novices who had worked fewer than 6 months with a computer system (n = 22). The second group consisted of those who had the most experience with their computer system (> 3 years with a

Table 6. Errors of Novices and Experts

Error Type	Mean No. of Errors/Hour Novices (n = 22)	Experts (n = 26)	t
Inefficiencies			
Inefficiency/habit	.26	.07	1.60
Inefficiency/knowledge	.81	.13	2.02*
Usability problems			
Knowledge base for regulation			
Knowledge errors	.92	.31	2.64**
Intellectual level of regulation			
Thought errors	.73	.17	1.94*
Memory errors	.28	.30	−.11
Judgement errors	.08	.05	.60
Level of flexible action patterns			
Habit errors	.17	.70	−2.49**
Omission errors	.72	.79	−.26
Recognition errors	.28	.19	.63
Sensorimotor level of regulation			
Sensorimotor errors	.61	.88	−.82
Functionality problems			
Action blockades	.09	.02	1.17
Action repetitions	.03	.07	−.66
Action interruptions	.08	.05	.31
Action detours	.12	1.43	−2.22*
Interaction problems[a]	.00	.19	

[a]Because there was no variance in the novice group, no t test could be computed.
*$p < .05$, one-tailed. **$p < .01$, one-tailed.

system; n = 26). Again, error frequencies per computer hour were compared. The results are presented in Table 6.

The results supported our hypothesis: Novices showed more knowledge errors, thought errors, and inefficient behavior due to knowledge than experts. On the other hand, there are no differences on the lower levels of regulation. However, there was one important exception: Most habit errors were committed by the expert group (by a ratio of 4:1). The distribution of action blockades, repetitions, and interruptions can hardly be interpreted, because the number of cases was too small. Interestingly, most action detours occurred in the expert group. One explanation is that experts use the systems up to their limits (Prümper et al., in press). Because of their expert knowledge they learned to deal with the weaknesses of their programs. In contrast, the novices probably try to suit their work tasks to the computer programs, or, if this is not possible, give up and change their goals.

5. OVERALL DISCUSSION

The goal of this article was to present an action-oriented error taxonomy and to show its construct validity. We shall first discuss the differentiations within each class of usability problems, functionality problems, and inefficiencies, then some observer problems, and finally, the practical implications will be described.

Usability problems are differentiated according to the steps in the action process and levels of regulation (with the additional group of knowledge errors). The differentiation of these error classes is warranted, when there are either empirical differences in the field data or when there are different practical suggestions related to the different error classes. Both reasons apply in the case of our taxonomy.

The taxonomy we have presented so far is able to integrate error taxonomies by Norman (1981), Rasmussen (1982, 1985, 1987a, 1987b), and Reason (1987a, 1990). Regarding usability problems, Hacker's (1986) levels of action regulation are very similar to Rasmussen's (1982, 1987b, 1987c) knowledge-based, rule-based, and skill-based levels. Reason (1987a) used this differentiation in his generic error-modelling system (GEMS). In our taxonomy there is an additional differentiation between the intellectual level and the knowledge base for regulation. We think that this differentiation is useful and has proved to have differential effects regarding complexity at work, computer expertise, and the need for external support. Theoretically and practically, there is a big difference between someone who does not know a certain command because of lack of knowledge and someone who is not able to build an adequate plan despite knowledge of the features of the computer system. However, we learned from our pilot study with computer experts (Zapf et al., 1989) that it is not always easy to differentiate between these categories empirically.

The intellectual level of regulation and the knowledge base for regulation were supposed to require more error-handling time and more external support. Knowledge errors should appear more frequently with novices; thought and memory errors should appear more often with highly complex jobs. These hypotheses were all confirmed in the observational field study.

Reason (1986) distinguished error classes of planning, storage, and execution. These error classes were considered to be correlated with mistakes, lapses, and slips. There is a high overlap of these error classes with our differentiation of the steps in the action process. In fact, Reason's (1986, 1987a) classification and Norman's (1981) activation-triggering-schema theory led us to revise a former version of our taxonomy (presented in Frese & Peters, 1988) and include memory and triggering processes into our theory. A similar error taxonomy using steps in the action process was also provided by Rouse and Rouse (1983).

Our hypotheses for the steps in the action process were not as explicit as for the levels of regulation, although there are important practical implications (as discussed later). Judgement errors can be differentiated from other errors on the intellectual level of regulation. They behaved similarly to knowledge errors, for example, in the amount of error-handling time and in the need for external support. There were also readily interpretable differences in the action steps on the level of flexible action patterns. Experts showed many more habit errors than novices, whereas this was not so for the other error types on the level of flexible action patterns. Additionally, recognition errors took longer to handle than other errors on the level of flexible action patterns. Thus, collapsing across the three error categories of habit, omission, and recognition, would not be useful.

Sensorimotor errors fall into a separate class. First, there was no need for support whatsoever. Second, they showed differences from habit errors in the distribution of novices and experts.

Functionality problems were similar to the usability problems of the higher levels of regulation in error-handling time. In the case of action detours, this was also true for need of support. However, there were different distributions of functionality problems (particularly action detours) and higher level usability errors for novices and experts. Action detours appeared only when a user knew how to compensate for a functionality problem in the system and when the limits of the system were pushed (Prümper et al., in press).

There were very few interaction problems in the study. Thus, we cannot be sure of this category yet. However, we assume that interaction problems will appear more frequently when networks are used more heavily.

The *practical implications* of our taxonomy are of major importance. A taxonomy is usually constructed with certain practical applications in mind. In our case, the taxonomy is supposed to help improve the construction of software programs and training. Because usability problems are the result of mismatches between program and person, practical suggestions can either improve training or the software.

There are two philosophies dealing with errors: error prevention and error management (Frese & Altmann, 1989; Frese, Irmer, & Prümper, 1991). Error prevention means to avoid errors as far as possible; this is the traditional strategy in software design and is particularly useful for functionality problems. They should be eradicated if possible. Users should be able to reach their goals easily with their computer programs and should not be additionally stressed by having to deal with functionality problems.

One general strategy to minimize errors is to provide system guidance. However, this reduces control, autonomy, and flexibility. System guidance may sometimes be useful and even necessary, for example, in nuclear power plants (cf. Rasmussen, 1988). But in the office, usually no catastrophes result from errors. For reasons of job design, autonomy and control at work are positive and should be enhanced. There are good reasons for software to provide control (Ackermann & Ulich, 1987; Frese, 1987a; Ulich, 1991; Ulich, Rauterberg, Moll, Greutmann, & Strohm, 1991). In contrast, the concept of *error management* allows control and still deals with the negative aspects of errors. The emphasis of this concept is not that the errors should be minimized, but their negative consequences (e.g., by quickly dealing with the errors).

Error management is useful because errors cannot be completely avoided anyhow, as the results of our study show. Errors do not disappear with experience. Even the experienced committed many usability errors, particularly at lower levels of regulation. As a matter of fact, the expert group committed even more habit errors than the novices (cf. Reason, 1987b). It is not possible to teach all the necessary features of a system because new tasks and new software features appear all the time. Errors can even have positive effects because they can be taken as a challenge to explore the system and improve the users' competence.

Thus, there should be more emphasis on error management. In training, errors should not just be avoided, but rather, strategies to handle errors better should be taught. Similarly, software should be designed to support error management. Therefore, there are four practical approaches to errors in human–computer interaction: support of error management either by training or by software, and reduction of errors by both software and training. For the sake of brevity, we will not give four examples for each error class (this was done in Frese, Brodbeck, et al., 1991; Irmer, Pfeffer, & Frese, 1991; Zapf, Frese, Irmer, & Brodbeck, 1991). A few examples will suffice to suggest the usefulness of the taxonomy.

For knowledge errors, training is obviously of particular relevance (Frese et al., 1988, Frese & Altmann, 1989). This is also suggested by the differences between novices and experts. Training to increase user knowledge should not only teach information about the system, but also about typical error situations and strategies or heuristics to deal with errors (Frese, Brodbeck et al., 1991; Greif, 1986; Greif & Keller, 1990). On the program side, knowledge errors point to the need to improve the "learnability" of the software. Learnability implies that there are obvious and consistent methods of representation (cf. the concept of mental models; Norman, 1986). Learnability is high when former knowledge can be transferred easily to a new system.

Thought errors point to the need for a higher transparency of a system (Maass, 1983). Transparency implies that the user can quickly and easily understand what the designer wanted to achieve. Transparent systems enhance the performance of the work task and thus support error prevention. As mentioned before, knowledge errors are more related to the interaction problem, whereas thought errors are more related to the task as a whole. Thought errors appear when the system does not support planning for doing the work tasks. For example, a spreadsheet task made it necessary to include many different variables. The user's plan did not include a variable necessary for the computation. Such a plan problem will probably appear more frequently if the system does not present all variables on the screen at the same time. Therefore, thought errors give a hint whether a system is easy to apply to one's tasks or not. The management of thought errors is supported when all relevant information is presented, when the system supports the reconstruction of the action course (e.g., history function), when it is easy to navigate in the system, and when it is possible to change and correct plans.

In terms of training, thought errors will appear less often if the training explicitly teaches "task-application knowledge." Task-application knowledge means that the users acquire knowledge of how to transfer the system knowledge to their real work situation (Papstein & Frese, 1988). In industry, many training programs do not include this feature and concentrate only on the basic system parameters.

Memory errors appear when parts of the action plan or information are forgotten. Like thought errors, this may be strongly related to the whole work task because they are due to the limits of the information-processing capacity. In training, strategies to reduce working-memory load can be taught (e.g., better naming strategies; cf. Carroll, 1982).

Management of memory errors is supported by systems that do not require keeping many bits of information in mind, but call them up from one action step to the next. However, such a system is hard to design. For example, we observed a system with four windows, all of them referring nicely to the tasks one-by-one. However, there were two problems. First, in some cases, results from a particular action step could not be moved from one window to the next. This meant that the users often had to input data twice. Second, there were problems with flexibility. For example, if a user was in the fourth window and wanted to correct an omission in the second one, it was not possible to go back. Rather, he or she had to start again. The management of memory errors is supported by the system, when it is always possible to interrupt the current action, when one can save the actual results, when one can go back to the point, where an action step was omitted, when it is easy to insert an omitted action, when it is easy to actualize data, and when it is easy to go back to the last action step.

Judgement errors refer to difficulties in interpreting system reactions. These kinds of errors could be avoided if system information were clear and easy to understand. Dean (1982), Lewis and Norman (1986), and Shneiderman (1987) gave suggestions on how to improve system messages. On the training side, general knowledge about the system would enhance the understanding of system reactions. However, many training programs mainly teach which keys to use rather than a general understanding of the system (Frese, 1987b, Frese et al., 1988). This is particularly true for computer tutorials (Caroll & Mazur, 1985; Greif & Janikowski, 1987); participants in these training situations have difficulties as soon as they get into unknown situations.

Our analyses showed that habit errors were frequently due to a lack of consistency between systems or between different parts in a system. Because routine actions do not require high conscious attention, actions are easily performed in the wrong situations (Lewis & Norman, 1986). For example, we observed a system that used several different words to confirm an action in different parts (return, execute, escape, "yes", enter). This led to problems for both novices and experienced users. The novices could not remember which key to press in which situation. The experienced users often pressed the wrong key habitually. These habit errors could be reduced by increasing the consistency and compatibility of the system (cf. Nielsen, 1989).

Sensorimotor errors occur within highly automatized actions. They can hardly be influenced by training. Regarding sensorimotor errors systems should be designed with high error robustness. This means that systems should tolerate small deviations from the correct input (cf. DIN, 1988). Because sensorimotor errors imply typos and mouse-movement errors, hardware design (keyboard, mouse, etc.) plays a major role here.

All of these suggestions for software design and training are preliminary and need further investigations. We think that our discussion shows, however, that an empirical taxonomy of errors produces theoretically interesting results and may be applied to improve both training and software design.

REFERENCES

Ackermann, D., & Ulich, E. (1987). The chances of individualization in human–computer interaction and its consequences. In M. Frese, E. Ulich, & W. Dzida (Eds.), *Psychological issues of human–computer interaction in the work place*. Amsterdam: North-Holland.

Adelson, B. (1984). When novices surpass experts: The difficulty of a task may increase with expertise. *Learning, Memory, and Cognition, 10*, 483–495.

Anderson, J.R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.

Arnold, B., & Roe, R. (1987). User errors in human–computer interaction. In M. Frese, E. Ulich, & W. Dzida (Eds.), *Psychological issues of human–computer interaction in the work place*. Amsterdam: North-Holland.

Bagnara, S., Stablum, F., Rizzo, A., Fontana, A., & Ruo, M. (1987, October). *Error detection and correction: A study on HCI in a hot strip mill production planning and control system*. Paper presented at the First European Meeting on Cognitive Science Approaches to Process Control. Marcoussis, France.

Bannon, L.J. (1986). Helping users help each other. In D.A. Norman & S.W. Draper (Eds.), *User-centered system design*. Hillsdale, NJ: Erlbaum.

Bons, H. (1985). Fehler und Fehlerauswertungen [Errors and error analysis]. In P. Gorny & W. Kilian (Eds.), *Computer-Software und Sachmängelhaftung*. Stuttgart: Teubner.

Booth, P.A. (1990). ECM: A scheme for analyzing user–system errors. In D. Diaper, D. Gilmore, G. Cockton, & B. Shackel (Eds.), *Human–computer interaction—INTERACT '90*. Amsterdam: Elsevier.

Bortz, J. (1984). *Lehrbuch der empirischen Forschung für Sozialwissenschaftler [Textbook on empirical research for social scientists]*. Berlin: Springer-Verlag.

Brodbeck, F.C., Prümper, J., & Zapf, D. (1988). Was denken Experten über Benutzerfehler? Phänomene, Ursachen and Behandlungsstrategien von Fehlern bei der Arbeit mit Büro-Software. [What do experts think about user errors? Phenomena, causes, and coping strategies of errors in working with office software] In Projekt FAUST (Ed.), *Zwischenbericht an der Projektträger Humanisierung des Arbeitslebens* (Vol. 2). Munich: University of Munich.

Brodbeck. F.C., Zapf, D., Prümper, J., & Frese, M. (1990). *Error handling in office work with computers: A field study*. Unpublished manuscript, University of Munich, Department of Psychology.

Carroll, J.M. (1982). Learning, using and designing file names and command paradigms. *Behavior and Information Technology, 1*, 327–346.

Carroll, J.M., & Mazur, S.A. (1985). *Lisa learning [RC report no. 11427]*. Yorktown Heights, NY: IBM Watson Research Center.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement, 20*, 37–46.

Cronbach, L.J., & Meehl, P.E. (1955). Construct validity in psychological tests. *Psychological Bulletin, 52*, 281–302.

Davis, R. (1983). Task analysis and user errors: A methodology for assessing interactions. *International Journal of Man–Machine Studies, 19*, 561–574.

Dean, M. (1982). How a computer should talk to people. *IBM System Journal, 21*, 424–453.

DIN. (1988). *Bildschirmarbeitsplätze. Grundsätze ergonomischer Dialoggestaltung [Workplaces with video display units. Basic principles of ergonomic dialog design] (DIN 66 234, Teil 8)*. Berlin: Beuth-Verlag.

Dörner, D. (1976). *Problemlösen als Informationsverarbeitungsprozesses* [Problem solving as information-processing]. Stuttgart: Kohlhammer.

Fleishman, E.A., & Quaintance, M.K. (1984). *Taxonomies of human performance*. London: Academic.

Frese, M. (1987a). A theory of control and complexity: Implications for software design and integration of the computer system into the workplace. In M. Frese, E. Ulich, & W. Dzida (Eds.), *Psychological issues of human–computer interaction at the workplace*. Amsterdam: North-Holland.

Frese, M. (1987b). The industrial and organizational psychology of human–computer interaction in the office. In C.L. Cooper & I.T. Robertson (Eds.), *International review of industrial and organizational psychology*. Chichester, England: Wiley.

Frese, M., Albrecht, K., Altmann, A., Lang, J., Papstein, P., 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., & Altmann, A. (1989). The treatment of errors in learning and training. In L. Brainbridge & A. Ruiz Quintanilla (Eds.), *Developing skills with information technology*. Chichester, England: Wiley.

Frese, M., Brodbeck, F.C., Heinbokel, T., Mooser, C., Schleiffenbaum, E., & Thieman, P. (1991). Errors in training computer skills: On the positive function of errors. *Human–Computer Interaction, 6,* 77–93.

Frese, M., Irmer, C., & Prümper, J. (1991). Das Konzept Fehlermanagement: Eine Strategie des Umgangs mit Handlungsfehlern in der Mensch-Computer Interaktion. [The concept of error management: A strategy for dealing with errors in human–computer interaction]. In M. Frese, Kasten, C., Skarpelis, & B. Zang-Scheucher (Eds.). *Software für die Arbeit von morgen. Bilanz und Perspektiven anwendungsorientierter Forschung*. Heidelberg: Springer.

Frese, M., & Peters, H. (1988). Zur Fehlerbehandlung in der Software-Ergonomie: Theoretische und praktische Überlegungen [Dealing with errors in software ergonomics: Theoretical and practical considerations]. *Zeitschrift für Arbeitswissenschaft, 42,* 9–17.

Frese, M., & Sabini, J. (Eds.). (1985). *Goal-directed behavior: The concept of action in psychology*. Hillsdale: Erlbaum.

Frese, M., & Stewart, J. (1984). Skill learning as a concept in life-span developmental psychology: An action-theoretic analysis. *Human Development, 27,* 147–162.

Frese, M., & Zapf, D. (in press). Action as the core of work psychology: A German approach. In M.D. Dunnette, L.M. Hough, & H. Triandis (Eds.), *Handbook of industrial and organizational psychology* (Vol. 4). Palo Alto, CA: Consulting Psychologists Press.

Gentner, D., & Stevens, A.L. (Eds.). (1983). *Mental models*. Hillsdale: Erlbaum.

Greif, S. (1986). Job design and computer training. *Bulletin of the British Psychological Society, 39,* 166–169.

Greif, S., & Janikowski, A. (1987). Aktives Lernen durch systematische Fehlerexploration oder programmiertes Lernen durch Tutorials [Active learning through systematic error exploration or programmed learning through tutorials]? *Zeitschrift für Arbeits- und Organisationspsychologie, 31,* 94–99.

Greif, S., & Keller, H. (1990). Innovation and the design of work and learning environments: The concept of exploration in human–computer interaction. In M.A. West & J.L. Farr (Eds.), *Innovation and creativity at work*. Chichester, England: Wiley.

Hacker, W. (1985). Activity: A fruitful concept in industrial psychology. In M. Frese & J. Sabini (Eds.), *Goal-directed behavior: The concept of action in psychology*. Hillsdale, NJ: Erlbaum.

Hacker, W. (1986). *Arbeitspsychologie* [Work psychology]. Bern: Huber.

Hacker, W. (1987). Computerization versus computer-aided mental work. In M. Frese, E. Ulich, & W. Dzida (Eds.), *Psychological issues of human–computer interaction in the workplace.* Amsterdam: North-Holland.

Hanson, S.J., Kraut, R.E., & Farber, J.M. (1984). Interface design and multivariate analysis of Unix command use. *ACM Transactions on Office Information Systems,* 2 (1), 42–57.

Heckhausen, H., & Beckmann, J. (1990). Intentional action and action slips. *Psychological Review, 97,* 36–48.

Herbrich, M., Frese, M., & Prümper, J. (1991).*Beobachterübereinstimmung bei der Analyse von Benutzerfehlern* [Observer agreement in user-error analysis]. Unpublished manuscript, University of Giessen, Department of Psychology.

Irmer, C., Pfeffer, S., & Frese, M. (1991). Konsequenzen von Fehleranalysen für das Training: Das Fehlertraining [Consequences of error analysis for training: Error training]. In M. Frese & D. Zapf (Eds.), *Fehler bei der Arbeit mit dem Computer. Ergebnisse von Beobachtungen und Befragungen im Bürobereich.* Bern: Huber.

Kraut, R.E., Hanson, S.J., & Farber, J.M. (1983). Command use and interface design. *Proceedings of CHI '83 Conference on Human Factors in Computing Systems,* 120–123.

Lang, T., Prümper, J., Wittmann, A., & Zapf, D. (1991). *Ein Experiment zum Fehlerbewältigungsspronzesses* [An experiment on the process of error handling] (Tech. rep.). Abschlussbericht des Projekt FAUST an den Projektträger Arbeit & Technik. Munich: University of Munich, Department of Psychology.

Lewis, C., & Norman, D.A. (1986). Designing for error. In D.A. Norman & S.W. Draper (Eds.), *User-centered system design.* Hillsdale, NJ: Erlbaum.

Maass, S. (1983). Why systems transparency? In T.R.G. Green, S.J. Payne, & G.C. van der Veer (Eds.), *The psychology of computer use.* London: Academic.

Miller, G.A., Galanter, E., & Pribram, K.H. (1960). *Plans and the structure of behavior.* London: Holt.

Nielsen, J. (1989). *Coordinating user interfaces for consistency.* Boston: Academic.

Nisbett, R.E., & Ross, L. (1980). *Human interference: Strategies and shortcomings of social judgement.* Englewood Cliffs, NJ: Prentice-Hall.

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

Norman, D.A. (1984). *Working papers on errors and error detection.* Unpublished manuscript, University of California, Institute for Cognitive Science, San Diego.

Norman, D.A. (1986). Cognitive engineering. In D.A. Norman & S.W. Draper (Eds.), *User-centered system design.* Hillsdale, NJ: Erlbaum.

Oesterreich, R. (1981). *Handlungsregulation und Kontrolle* [Action regulation and control]. Munich: Urban & Schwarzenberg.

Oesterreich, R., & Volpert, W. (1986). Task analysis for work design on the basis of action regulation theory. *Economic and Industrial Democracy, 7,* 503–527.

O'Malley, C.E. (1986). Helping users help themselves. In D.A. Norman & S.W. Draper (Eds.), *User centered system design.* Hillsdale, NJ: Erlbaum.

Papstein, P.V., & Frese, M. (1988). Transferring skills from training to the actual work situation: The role of application knowledge, action styles and job decision latitude. In E. Soloway, D. Frye, & S.B. Sheppard (Eds.), *Proceedings of the CHI '88 conference of human factors in computing systems.* Reading, MA: Addison-Wesley.

Peters, H., Frese, M., & Zapf, D. (1990). Nutzungs- und Funktionsprobleme bei unterschiedlichen Dialogsystemen [Usability and functionality problems in different dialog systems]. *Zeitschrift für Arbeitswissenschaft, 44,* 145–152.

Prümper, J. (1991). Die Inter-Rater-Reliabilität von Fehlerbeobachtungen im Feld [Interrater reliability of error observation in the field]. In M. Frese & D. Zapf (Eds.), *Fehler bei der Arbeit am Computer. Ergebnisse von Beobachtungen und Befragungen im Bürobereich.* Bern: Huber.

Prümper, J. (in press). *Methodische Probleme bei der Erfassung von Fehlern in der Mensch–Computer Interaktion am Arbeitsplatz.* [Methodical problems during error recording in human–computer interaction at the workplace] Unpublished dissertation thesis, University of Giessen.

Prümper, J., Zapf, D., Brodbeck, F.C., & Frese, M. (in press). Errors of novices and experts: Some surprising differences from computerized office work. *Behaviour and Information Technology.*

Rasmussen, J. (1982). Human errors: A taxonomy for describing human malfunction in industrial installations. *Journal of Occupational Accidents, 4,* 311–335.

Rasmussen, J. (1985). *Human error data. Facts or fiction.* Roskilde, Denmark: Riso National Laboratory.

Rasmussen, J. (1987a). Cognitive control and human error mechanisms. In J. Rasmussen, K. Duncan, & J. Leplat (Eds.), *New technology and human error.* Chichester, England: Wiley.

Rasmussen, J. (1987b). Reasons, causes, and human error. In J. Rasmussen, K. Duncan, & J. Leplat (Eds.), *New technology and human error.* Chichester, England: Wiley.

Rasmussen, J. (1987c). The definition of human error and a taxonomy for technical system design. In J. Rasmussen, K. Duncan, & J. Leplat (Eds.), *New technology and human error.* Chichester, England: Wiley.

Rasmussen, J. (1988). Human error mechanisms in complex work environments. *Reliability Engineering and System Safety, 22,* 155–167.

Reason, J. (1986, December). *Intentions, errors, and machines: A cognitive science perspective.* Paper presented at the Aspects of Conciousness and Awareness conference. Bielefeld, Germany.

Reason, J. (1987a). Generic error-modelling system (GEMS): A cognitive framework for locating common human error forms. In J. Rasmussen, K. Duncan, & J. Leplat (Eds.), *New technology and human error.* Chichester, England: Wiley.

Reason, J. (1987b). The cognitive bases of predictable human error. *Proceedings of the Ergonomics Society's Annual Conference.*

Reason, J. (1990). *Human error.* New York: Cambridge University Press.

Rizzo, A., Bagnara, S., & Visciola, M. (1987). Human error detection processes. *International Journal of Man–Machine Studies, 27,* 555–570.

Rouse, W.B., & Rouse, S.H. (1983). Analysis and classification of human error. *IEEE Transactions on Systems, Man, and Cybernetics, 13,* 539–549.

Scharer, L.L. (1983). User training: Less is more. *Datamation, 29,* 175–182.

Schmidt, R.A. (1975). A schema theory of discrete motor skill learning. *Psychological Review, 82,* 225–260.

Schönpflug, W. (1985). Goal-directed behavior as a source of stress: Psychological origins and consequences of inefficiency. In M. Frese & J. Sabini (Eds.), *Goal-directed behavior: The concept of action theory in psychology.* Hillsdale, NJ: Erlbaum.

Sellen, A.J. (1990). *Four chapters on human error and human error detection.* Unpublished dissertation thesis, Institute for Cognitive Science, University of California, San Diego.

Semmer, N. (1984). *Stressbezogene Tätigkeitsanalyse* [Stress oriented job analysis]. Weinheim: Beltz.

Semmer, N., & Frese, M. (1985). Action theory in clinical psychology. In M. Frese & J. Sabini (Eds.), *Goal-directed behavior: The concept of action in psychology.* Hillsdale, NJ: Erlbaum.

Shneiderman, B. (1987). *Designing the user interface: Strategies for effective human–computer interaction.* Reading, MA: Addison-Wesley.

Smelcer, J.B. (1989). *Understanding user errors in database query.* Unpublished dissertation thesis, University of Michigan, Ann Arbor.

Ulich, E. (1989). Arbeitspsychologische Konzepte der Aufgabengestaltung [Work psychology concepts of task design]. In S. Maass & H. Oberquelle (Eds.), *Software-Ergonomie '89. Aufgabenorientierte Systemgestaltung.* Stuttgart: Teubner.

Ulich, E. (1991). *Arbeitspsychologie* [Work psychology]. Stuttgart: Poeschel.

Ulich, E., Rauterberg, M., Moll, T., Greutmann, T., & Strohm, O. (1991). Task orientation and user-oriented dialog design. *International Journal of Human–Computer Interaction, 3,* 117–144.

Volpert, W. (1982). The model of the hierarchical-sequential organization of action. In W. Hacker, W. Volpert, & M.V. Cranach (Eds.), *Cognitive and motivational aspects of action.* Berlin: Deutscher Verlag der Wissenschaften.

Volpert, W. (1987). Psychische Regulation von Arbeitstätigkeiten [Psychological regulation of work activities]. In U. Kleinbeck & J. Rutenfranz (Eds.), Enzyklopädie der Psychologie, Themenbereich D, Serie III, Band 1: *Arbeitspsychologie.* Göttingen: Hogrefe.

Volpert, W.V. Oesterreich, R., Gablenz-Kolakovic, S.I. Krogoll, T., & Resch, M. (1983). *Verfahren zur Ermittlung von Regulationserfordernissen in der Arbeitstätigkeit (VERA)* [Instrument to identify regulation requirements in industrial work]. Köln: TÜV-Rheinland.

Zapf, D. (1989). *Selbst- und Fremdbeobachtung in der psychologischen Arbeitsanalyse. Methodische Probleme bei der Erfassung von Stress am Arbeitsplatz* [Self- and outside observation in psychological work analysis. Methodical problems of measuring stress in the workplace]. Göttingen: Hogrefe.

Zapf, D. (1991). Stressbezogene Arbeitsanalyse bei der Arbeit mit unterschiedlichen Bürosoftwaresystemen [Stress oriented job analysis at work places with different software systems]. *Zeitschrift für Arbeits- und Organisationspsychologie, 35,* 2–14.

Zapf, D., Brodbeck, F.C., & Prümper, J. (1989). Handlungsorientierte Fehlertaxonomie in der Mensch–Computer Interaktion. Theoretische Überlegungen und eine erste Überprüfung im Rahmen einer Expertenbefragung. [An action-oriented error taxonomy in human–computer interaction. Theoretical considerations and a first empirical examination in the course of an expert survey]. *Zeitschrift für Arbeits- und Organisationspsychologie,* 33, 178–187.

Zapf, D., & Frese, M. (1989). Benutzerfehler im Kontext von Arbeitsaufgabe und Arbeitsorganisation [User error in the context of work task and work organization]. In S. Maass & H. Oberquelle (Eds.), *Software Ergonomie '89. Aufgabenorientierte Systemgestaltung.* Stuttgart: Teubner.

Zapf, D., Frese, M., Irmer, C., & Brodbeck, F.C. (1991). Konsequenzen von Fehleranalysen für die Softwaregestaltung [Consequences of error analysis for software design]. In M. Frese & D. Zapf (Eds.), *Fehler bei der Arbeit mit dem Computer: Ergebnisse von Beobachtungen und Befragungen im Bürobereich.* Bern: Huber.

Zapf, D., Lang, T., & Wittmann, A. (1991). Der Fehlerbewältigungsprozess [The process of error handling]. In M. Frese & D. Zapf (Eds.), *Fehler bei der Arbeit mit dem Computer. Ergebnisse von Beobachtungen und Befragungen im Bürobereich.* Bern: Huber.

Quelle:
Zapf, D., Brodbeck, F.C., Frese, M., Peters, H. & Prümper, J. (1992). Errors in working with office computers. A first validation of a taxonomy for observed errors in a field setting. *International Journal of Human-Computer Interaction, 4,* 311-339.