To Err Or Not To Err, That Is The Question: Novice User Perception of Errors While Surfing The Web

Jonathan K. Lazar
Department of Information Systems, University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250
Phone 410-455-2851, Fax 410-455-1073, jlazar1@umbc.edu

Anthony F. Norcio
Department of Information Systems, University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250
Phone 410-455-3938, Fax 410-455-1073, norcio@umbc.edu

ABSTRACT

Novice users frequently make errors as they attempt to perform new tasks. These errors tend to frustrate novice users. In the networked environment, there is an increased opportunity for users to perceive that they make errors. Much of the research on errors was done before the networked environment was widespread, and the research does not address this new task environment. This paper introduces the concept of a "situational error." A situational error is when a user has entered their commands correctly to reach a goal, but the user is not able to reach their goal due to outside factors such as network bottlenecks and remote site failures. Situational errors are described in detail, and are placed into the traditional taxonomies of errors. Methods for designing systems and training to assist novice users in responding to errors are also presented.

INTRODUCTION

As users learn to use a new application of computer technology, they frequently make errors (Greif and Keller, 1990; Lazonder and Meij, 1995; Norman, 1983). These errors tend to frustrate users (Arnold and Roe, 1987; Carroll and Mack, 1984). There is an extensive body of literature classifying the different types of errors that novice users make when interacting with computer systems. However, much of this literature was created before the networked environment and the Internet were prevalent. The current literature does not address many of the user-perceived errors that can occur in the networked environment. The purpose of this paper is to introduce the concept of a "situational error."

SITUATIONAL ERRORS

When users are connected to a network, whether it is a local area network, or the Internet, they have task goals in mind. They may want to print a document, retrieve a file from the Internet, or access the latest news from a web site. Novice users may have the "traditional system" mindset, which is that, if they enter their commands in a correct manner, then they will be able to reach their goal. However, in the networked environment, this is not always the case. A user's task goal may require the use of a network resource which is not available or is not functioning properly. Even if the user enters all of their commands in the appropriate manner, they may not be able to reach their task goal. We have named this a "situational error." A situational error is when a user's task goal requires the use of a network resource, which is not available or is not functioning properly.

Why would a situational error occur? There may be a problem with the connection from a user's computer to a local area network. There may be a problem with the connection between a local network and an Internet Service Provider. Domain name service may not be set up correctly, so users on a local network may not be able to access sites on the world wide web. A network resource may not be properly configured. A remote site may have failed, or there may be a bottleneck somewhere on the network (Johnson, 1998). Regardless of why users could not reach their goal, many novice users do not have a good understanding of the technologies and protocols that support their network, and therefore, cannot understand when they do not meet their goals (Johnson, 1998). Therefore, the user

may view this occurrence as an error. Should this even be considered an error? This depends on the working definition of error. The next section discusses the various definitions of errors.

CLASSIFICATIONS OF ERROR DEFINITIONS

Within the literature, several different definitions of errors have been proposed. To understand errors, it is important to examine the many definitions of error. Arnold and Roe point out that within error definitions, there are two very different views of errors (Arnold and Roe, 1987). One set of error definitions is *user-centered*; the other set of definitions is *system-centered* (Arnold and Roe, 1987).

User-centered definition of error

Users want to complete their tasks successfully. User-centered definitions consider errors from the point of view of the user. User-centered definitions of error view an error as when a user's desired action is not carried out (Norman, 1983). Users are concerned with reaching their goals, and from the users' point of view, errors keep them from reaching those goals. Some of the user-centered definitions of error that have been presented in the literature are "when a user's intention or goal is not attained," (Arnold and Roe, 1987, p. 204) and "the non-attainment of a goal" (Frese and Altmann, 1989).

User-centered definitions do not blame users for errors. User-centered definitions of error only state that errors keep users from reaching their goals. Zapf et. al. point out that for an error to be defined as such, a specific program or system must be designed to perform the task that the users want (Zapf et al., 1992). If the user has a specific goal, but the program or system is not designed to perform the tasks to reach such a goal, then this is called a functionality problem (Goodwin, 1987). For instance, if a user attempts to use a statistics program to browse the web, this would be considered a functionality problem, not an error, because the application (the statistics program) was not designed to meet the user's task goal (browsing the web). According to these classifications, a situational error is still an error, since functionality is not a problem. The applications and systems are designed to perform the tasks to reach the users' goals.

System-centered definitions of error

System-centered definitions of error view errors from the system's point of view; user goals are not addressed. System-centered definitions of error are more technically-oriented. From the system's point of view, if something cannot process successfully, it is due to an error on the user's part. System-centered definitions of error blame the users (Lewis and Norman, 1986). Some of the system-centered definitions of error in the literature include:

- An action that violates a rule (Frese and Altmann, 1989).
- Something that the system cannot respond to (Lewis and Norman, 1986, p. 411).
- Actions that are inappropriate (Booth, 1991).

Although system-centered definitions of error blame the user for errors, it is pointless to blame a user (Zapf et al., 1992). In human-computer interaction research, the focus is on assisting and designing for users (Dix et al., 1998; Preece et al., 1994; Shneiderman, 1998). Assigning blame doesn't help the user. Instead, it is important to focus on assisting the user in reaching their goal.

USER PERCEPTION OF ERROR

How do users perceive these situational errors? Do users perceive these errors are their own fault? Or do users perceive them as someone else's fault? With traditional classifications of errors, novice users tend to blame themselves for making an error (Carroll and Mack, 1984; Lewis and Norman, 1986). Errors intimidate novice users more than expert users, who are confident in their abilities (Carroll, 1990). Expert users tend to blame anyone or anything else (the program, the manual, the system designer) before they blame themselves for an error (Carroll, 1990). Expert users may not even consider a situational error an error at all. Expert users are used to dealing with errors and other odd situations, and may just consider these errors as par for the course (Somekh and Davis, 1997). Will these same patterns of blame appear with situational errors? Experimental work needs to be done with users and situational errors.

When a situational error occurs, the user has taken an action, by performing a set of commands, in the appropriate manner. The user then expects to get the appropriate result. However, through no fault of their own, the user cannot achieve their goal. What may frustrate the user even more is the fact that they have performed the same actions previously, with a successful result. The user may wonder why, in this specific situation, is the same set of commands not producing the same result? This can be confusing to the user.

TRADITIONAL CLASSIFICATIONS OF ERRORS

How do situational errors fit into the traditional classifications of errors? The classification system for user

errors that is prominent in the HCI world is provided by Donald Norman. At the highest level, Norman separates errors into two types: mistakes and slips (Norman, 1983). Norman defines a mistake as when users choose the wrong commands to reach their goals (Arnold and Roe, 1987; Norman, 1983). A mistake has also been called a conceptual error (Booth, 1991). A slip, on the other hand, is when a user's intended command is correct, but the user makes an error (such as a spelling error) in entering their commands (Norman, 1983). Within slips, there are many different classifications. For instance, Norman defines mode errors, description errors, capture errors, activation errors, and data description errors.

Mode errors

Frequently, applications and systems have different modes. A keypress while the system is in one mode will provide a different action then the same keypress when the system is in a different mode. A mode error is when users believe that a system is in one mode, when instead, it is in another mode (Norman, 1988). Users then perform their actions with the mistaken belief that the system is in a certain mode. Because the system is in a different mode, their actions may have results other than what the users intended, and user does not reach their goals.

Description errors

Many times, different actions or procedures are carried out using a similar set of commands. A description error is when users perform a procedure in a correct manner, but perform it on the wrong file, item or object (Norman, 1983; Norman, 1988). An example of a description error could be to send a business colleague the wrong file as an e-mail attachment. The user performed the procedure in the correct manner, saving a file in the correct directory, uploading it to their e-mail account, and correctly attaching it to their e-mail message. However, these procedures were performed on the wrong file. Description errors frequently occur when objects, files, or items, look similar or are physically close to each other.

Capture errors

A capture error is when there is overlap between one set of commands and another, and the user performed the wrong sequence of commands (Norman, 1983). When attempting to perform one set of commands, another set of commands, which is similar, "takes over" (Norman, 1988). There are many commands that are similar. In some of these cases, the commands partially overlap. For instance, a keystroke sequence of control-alt-delete reboots most personal computers. If the key sequence control-alt by itself performed a procedure, it is expected that many times, users would mean to type only control-alt, but instead would type control-alt-delete.

Activation errors

An activation error occurs when users fail to complete all of the required procedures to reach their intention (Norman, 1983). This may be due to other events that have occurred while users are performing the appropriate actions. These other events take away the users' attention, causing them to forget the exact procedures to execute and the order in which the procedures should be executed (Norman, 1983). Norman later renamed this specific type of error as a "loss-of-activation" error (Norman, 1988).

Data-driven errors

A data-driven error is based on the arrival of data to our senses (Norman, 1988). Users may be in the process of entering data, when someone tells them that the current ballgame score is 3 to 1, Orioles winning. The users then may enter the numbers 31 instead of the actual data that should be entered. This differs from an activation error, because it does not completely end the users' procedure. The user can continue with the procedure, however, they will have entered incorrect data at an earlier point.

Situational errors

A situational error does not fit into any of these traditional classifications of errors. A situational error cannot be considered a slip. Since the user does not carry out the commands incorrectly, a situational error is not a slip. Furthermore, since the user has chosen the correct commands to reach their goals, a situational error is not a mistake. In mistakes and slips, it is assumed that the user caused an error. However, there is no classification for when a user selected the correct commands, and entered the commands in the correct manner, but was still not able to attain their goal. Other error classification schema have been presented in the literature (Zapf et al., 1992), but they, too, do not address these situational errors.

DESIGNING FOR SITUATIONAL ERRORS

How can system designers and trainers assist novice users in responding to situational errors? There are two

possible approaches to assisting users in responding to errors: system design and training design.

System design

Systems should be designed in a way that they assist the users in responding to errors. For instance, Lazonder and Meij state that users need to be aware of the error as quickly as possible, so that they can attempt to correct it (Lazonder and Meij, 1995). If users are not immediately aware of an error, the error can be compounded over time, into a larger error which is harder to recover from (Carroll and Carrithers, 1984; Carroll and Mack, 1984). Lazonder and Meij also emphasize the location of the error message. If an error message is displayed in a dialog box superimposed on the center of the active screen, users will be likely to notice it (Lazonder and Meij, 1995). However, if an error message is displayed in the lower right hand corner of the screen, and no other signals (graphics, sounds, etc.) draw the user's attention to the error message, the user might not even notice the error message (Lazonder and Meij, 1995).

The error messages themselves should be designed in a way that is clear and easy for the user to understand. Both Brown and Shneiderman emphasize the importance of easy-to-understand error messages (Brown, 1983; Shneiderman, 1998). A bad example of an error message, as described by DuBoulay and Matthew, is the error message "fatal error in pass zero" (DuBoulay and Matthew, 1984). In their anecdotal experience with students, novice users have had trouble understanding the meaning of that message (DuBoulay and Matthew, 1984).

In an experiment with 22 novice programmers, Shneiderman found that error messages that are easier for the user to understand and interpret can result in users being better able to respond appropriately to errors (Shneiderman, 1982). Shneiderman suggests that error messages should 1) be specific, 2) be positive, and 3) tell users what to do to respond to the error (Shneiderman, 1998). In their conceptual paper, Arnold and Roe go one step further, by encouraging system designers to not just tell the users what to do, but to give users information about the different alternatives that they have to respond to the error (Arnold and Roe, 1987). This can facilitate recovery from the error.

To assist novice users in responding to situational errors, more information needs to be provided to the user. What is Domain Name Service? What is a "404" error? The novice user may not know what these things are. Therefore, it may be necessary to provide the novice user with further information, such as a message saying "There is an error on the network, but it is not due to your actions." Another approach, such as "The network is experiencing problems; please try again later" might be appropriate. This is similar to the phone company message "All circuits are busy. Please try again later." These error messages let the novice user know that, although they may not be immediately able to reach their task goal, that 1) this is not their fault and 2) they should attempt their goal again later that day.

Training design

Since users may perceive that there are many more errors when learning to use the Internet, user training should focus on assisting users in responding to these errors. Alternative approaches for training have been suggested, to help novice users deal with errors (Dormann and Frese, 1994; Frese and Altmann, 1989; Frese et al., 1991; Nordstrom, Wendland and Williams, 1998). Error management training, introduced by Frese and Altmann in 1989, focuses on turning errors into positive opportunities for learning (Frese and Altmann, 1989). Users are given a set of "error heuristics," telling them that errors are not bad, but that errors are good opportunities for learning, and that they can figure out an appropriate response to the error. Another approach is exploration, in which users are given an overview of their task environment, and how to navigate through it (Carroll and Mack, 1984; Carroll and Mazur, 1986; Frese and Altmann, 1989). This replaces the traditional training method, where users are given a specific list of exactly what to type in order to reach their task goal. By familiarizing users with the structure of their environment, exploratory training might assist users in understanding why they weren't able to reach their task goals. Conceptual models may also assist users in gaining a better understanding of the structure of their task environment (Santhanam and Sein, 1994; Sein, Bostrom and Olfman, 1987). Conceptual models are graphical or mathematical representations of a system that correspond closely to the real-world system (Santhanam and Sein, 1994). With a better understanding of their task environment, users might not become as frustrated with situational errors, since they would understand the cause of their errors. Since errors are more prevalent in the networked environment, it is important to train novice users to better respond to errors.

SUMMARY

This paper serves as an introduction to the concept of situational errors. Errors can frustrate novice users, and in the networked environment, there is an increased opportunity for users to make the form of error known as a situational error. It is important to understand what a situational error is, and know how to design systems and training to assist users in responding to errors. Experimental work with users is needed to learn more about situational errors and how users are affected by them.

REFERENCES

Arnold, B., and Roe, R. (1987). User errors in human-computer interaction. In M. Frese, E. Ulich, and W. Dzida (Eds.), Human computer interaction in the workplace (203-220). Amsterdam: Elsevier Science Publishers.

Booth, P. (1991). Errors and theory in human-computer interaction. Acta Psychologica, 78(1/3), 69-96.

Brown, P. (1983). Error messages: The neglected area of the man/machine interface. Communications of the ACM, 26(4), 246-249.

Carroll, J. (1990). The nurnberg funnel: Designing minimalist instruction for practical computer skill. Cambridge, Massachusetts: MIT Press.

Carroll, J., and Carrithers, C. (1984). Training wheels in a user interface. Communications of the ACM, 27(8), 800-806.

Carroll, J., and Mack, R. (1984). Learning to use a word processor: By doing, by thinking, and by knowing. In J. Thomas, and M. Schneider (Eds.), Human Factors in Computer Systems (13-51). Norwood, N.J.: Ablex Publishing.

Carroll, J., and Mazur, S. (1986). LisaLearning. IEEE Computer, 19(11), 35-49.

Dix, A., Finlay, J., Abowd, G., and Beale, R. (1998). Human-Computer Interaction. (2nd ed.). London: Prentice Hall England.

Dormann, T., and Frese, M. (1994). Error training: Replication and the function of exploratory behavior. International Journal of Human-Computer Interaction, 6(4), 365-372.

DuBoulay, B., and Matthew, I. (1984). Fatal error in pass zero: How not to confuse novices. Behaviour and Information Technology, 3(2), 109-118.

Frese, M., and Altmann, A. (1989). The treatment of errors in learning and training. In L. Bainbridge, and S. Quintanilla (Eds.), Developing skills with information technology (65-86). Chichester, England: John Wiley & Sons.

Frese, M., Brodbeck, F., Heinbokel, T., Mooser, C., Schleiffenbaum, E., and Thiemann, P. (1991). Errors in training computer skills: On the positive function of errors. Human-Computer Interaction, 6(1), 77-93.

Goodwin, N. (1987). Functionality and usability. Communications of the ACM, 30(3), 229-233.

Greif, S., and Keller, H. (1990). Innovation and the design of work and learning environments: The concept of exploration in human-computer interaction. In M. West, and J. Farr (Eds.), Innovation and creativity at work: Psychological and organizational strategies (231-249). Chichester, England: John Wiley & Sons.

Johnson, C. (1998). Electronic gridlock, information saturation, and the unpredictability of information retrieval over the world wide web. In P. Palanque, and F. Paterno (Eds.), Formal Methods in Human-Computer Interaction (261-282). London: Springer.

Lazonder, A., and Meij, H. (1995). Error-information in tutorial documentation: Supporting users' errors to faciliate initial skill learning. International Journal of Human-Computer Studies, 42(2), 185-206.

Lewis, C., and Norman, D. (1986). Designing for error. In D. Norman, and S. Draper (Eds.), User-centered system design (411-432). Hillsdale, NJ: Lawrence Erlbaum Associates.

Nordstrom, C., Wendland, D., and Williams, K. (1998). To err is human: An examination of the effectiveness of error management training. Journal of Business and Psychology, 12(3), 269-282.

Norman, D. (1983). Design rules based on analyses of human error. Communications of the ACM, 26(4), 254-258.

Norman, D. (1988). The psychology of everyday things: Harper Collins Publishers.

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., and Carey, T. (1994). Human-Computer Interaction. Wokingham, England: Addison Wesley Publishing.

Santhanam, R., and Sein, M. (1994). Improving end-user proficiency: Effects of conceptual training and nature of interaction. Information Systems Research, 5(4), 378-399.

Sein, M., Bostrom, R., and Olfman, L. (1987). Conceptual models in training novice users. Proceedings of the Human-Computer Interaction- INTERACT '87, Stuttgart, Germany; 861-867.

Shneiderman, B. (1982). System message design: Guidelines and experimental results. In A. Badre, and B. Shneiderman (Eds.), Directions in Human/Computer Interaction (55-78). Norwood, N.J.: Ablex Publishing.

Shneiderman, B. (1998). Designing the User Interface: Strategies for Effective Human-Computer Interaction. (3rd ed.). Reading, Massachusetts: Addison-Wesley.

Somekh, B., and Davis, N. (1997). Getting teachers started with IT and transferable skills. In B. Somekh, and N. Davis (Eds.), Using information technology effectively in teaching and learning (138-149). London: Routledge.

Zapf, D., Brodbeck, F., Frese, M., Peters, H., and Prumper, 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(4), 311-339.