Understanding and Assisting CAD Users in the Real World

by

Suresh K. Bhavnani1
Carnegie Mellon University
Engineering Design Research Center (EDRC)
Pittsburgh, Pennsylvania

&

Ulrich Flemming
Carnegie Mellon University
Engineering Design Research Center (EDRC)
Pittsburgh, Pennsylvania

&

Diana E. Forsythe
Intelligent Systems Laboratory
Department of Computer Science
University of Pittsburgh
Pittsburgh, Pennsylvania

&

James H.Garrett, Jr.
Carnegie Mellon University
Engineering Design Research Center (EDRC)
Pittsburgh, Pennsylvania

&

Doris S. Shaw
Construction Engineering Research Laboratory (CERL)
U.S. Army Corps of Engineers
Champaign, Illinois

Suresh Bhavnani

is a Ph.D. student in the Department of Architecture at Carnegie Mellon University. A former employee of the Intergraph Corporation, he is interested in improving the productivity of CAD systems by studying usability issues from an ethnographic as well as a cognitive perspective. His doctoral thesis involves the nature and prediction of CAD interaction errors.

Ulrich Flemming

is a professor in the Department of Architecture at Carnegie Mellon University. He obtained a M.Arch degree from MIT and a Ph.D. from the Technical University of Berlin. His research areas include theories of form generation in architecture, computer-aided design, knowledge-based design, case-based design, design databases, integrated design systems, distributed collaboration in design.

Diana E. Forsythe

is a cultural anthropologist. She obtained a B.A. from Swarthmore and a Ph.D. from Cornell, and did post-doctoral research in artificial intelligence at Stanford. Formerly Research Associate Professor of Computer Science and Anthropology at the University of Pittsburgh, she is currently a System Development Foundation Fellow at Stanford.

James H. Garrett, Jr.

is an associate professor in the Department of Civil and Environmental Engineering at Carnegie Mellon University. His research and teaching interests are oriented toward the application of computing technology to the engineering tasks of analysis, synthesis and evaluation. He has been investigating the usage of object-oriented programming for representing engineering design standards and the usage of neural networks for solving problems in structural and environmental engineering.

Doris Smith Shaw

is a principal investigator at the U. S. Army Construction Engineering Research Laboratories in Champaign, Illinois. Her main technical and research interest is intelligent performance support systems for Architectural and Engineering CAD applications. She received her Ph.D. from the University of Illinois in Educational Technology.

Understanding and Assisting CAD Users in the Real World

Abstract

In spite of the rapid increase in functionality and resources provided by CAD systems, productivity growth expected from their use has been difficult to achieve. Although many surveys describe this "productivity puzzle", few studies have been conducted on actual CAD users to understand its causes. In an effort to understand this issue, the first author visited a federal architectural office and observed CAD users in their natural setting using ethnographic techniques developed by cultural anthropologists. This paper describes preliminary results obtained from the study. The study revealed that users had leveled-off in their learning and experimentation and were using the CAD system in sub-optimal ways. The authors argue that this sub-optimal usage occurs because users have limited ways to learn better or different ways of executing tasks. The authors propose that CAD systems should provide active assistance, that is, intervene spontaneously with advice, assistance, and relevant information while the user interacts with the system. They conclude with some issues revealed by the study that should be considered when developing such active assistance.

Introduction

Productivity increases through the use of computers have been negligible or difficult to achieve in various application domains. The huge investments in the computer revolution, in general, have not paid off in terms of productivity growth, a phenomenon that is commonly referred to as the 'productivity puzzle' (Forester, 1989). Productivity in firms using CAD systems does not differ much from this general picture. Firms that have used their system for one year report productivity increases of only 5% and typically do not achieve the maximum productivity growth until they have worked with CAD for five years (PSMJ CADD Applications and User Survey, 1994).

The few laboratory and field studies on CAD usage that are available present a dismal picture. Bietz et al. (1990) found that mechanical engineering students who had passed a CAD course produce better and more complete drawings with less effort using paper and pencil than on a CAD system. Luczak et al. (1991) studied 43 subjects using 11 CAD systems in 11 factories. They found that even when the subjects were highly trained, the high complexity of the commands (due to many input parameters, restrictions, and requirements) led to low performance, reduced creativity, frictions, and frustrations. Finally, Majchrzak (1990) found no improvement in the performance of 25 engineers and 60 drafters using CAD systems in comparison to non-CAD users.

Many reasons have been offered for these disappointing results. These include blaming the users for not reading manuals, not using help, not getting adequate training and not modifying their work process to use the new technology intelligently. Others blame the

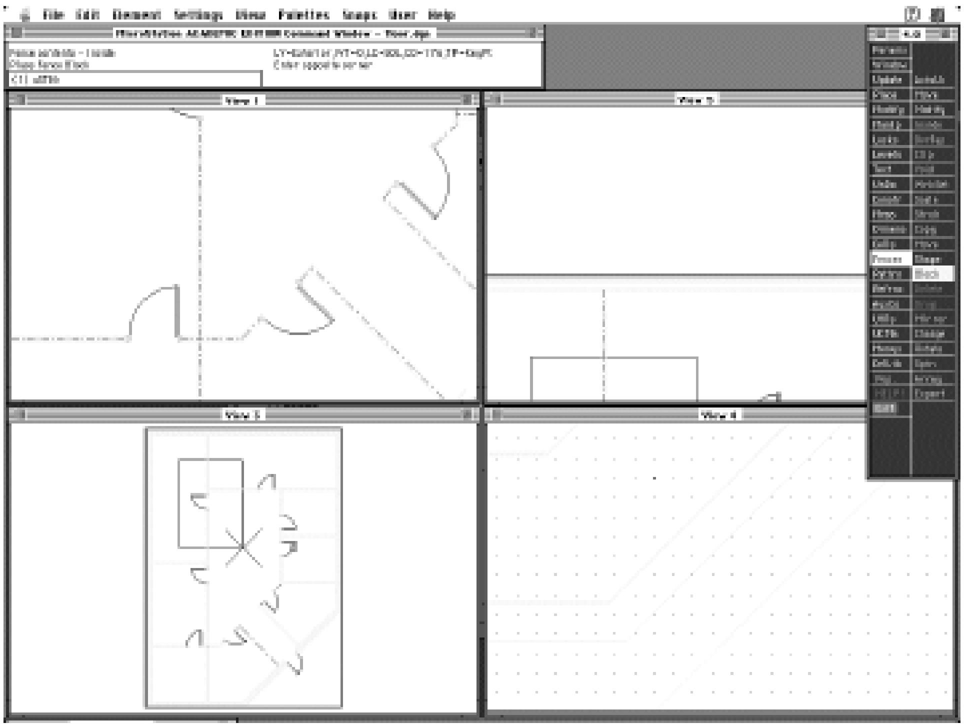

Figure 1. MicroStation interface showing typical window and menu set-up.

CAD system for having poorly designed and unnatural interfaces, non-adaptive interfaces, inadequate and unstable functionality, and poorly designed help, training, and documentation.

We were intrigued by the productivity puzzle and the possible reasons behind it. Having a wide range of research tools and technologies at our disposal, we asked how they could be used to improve the usability of CAD systems, particularly for architects. We decided to start by observing real users doing CAD work in their natural environment. We hoped that an in-depth case study would provide us with first-hand observations of successes and failures in CAD usage. This paper describes preliminary findings from this study of end-users and offers some suggestions for future system design.

Observing People in Real-World Settings

Having decided to study architectural CAD users in their natural environment, we investigated techniques that would be most appropriate for such a study. Cultural anthropologists have developed techniques to observe people in real-world settings (Powdermaker, 1966; Werner and Schoepfle, 1987). Their method is known as ethnography. Originally applied in non-Western societies, ethnographic techniques have been used more recently to illuminate computer usability problems (Forsythe, 1992; Lundsgaarde, 1987; Nardi, 1993; Suchman, 1987) . For example, Forsythe demonstrates how ethnography can be used to help design a computer-based explanation system for migraine sufferers (Forsythe in press). The study included extensive observation of interactions between doctors and patients in clinical settings as well as in-depth interviews with people with migraine.

The main ethnographic data-gathering method, known as participant observation, involves the use of unobtrusive observational techniques. Building rapport with users, trained fieldworkers systematically immerse themselves in the users' work environment in order to provide qualitative feedback to designers. Such immersion is intended to provide information not only about users' work practice, but also about their point of view, organizational setting, social interactions, values and assumptions. Such understanding aids the fieldworker in interpreting observational data. Participant observation may be supplemented as appropriate by interviewing, by active intervention, and by a range of formal and informal techniques for the elicitation and recording of quantitative as well as qualitative data.

The hallmark of this research method is its flexibility. Adaptable to a wide range of real-world settings, it allows the fieldworker to refine and modify a research question over time as preconceived notions of user needs are replaced by a developing understanding of what the real issues are from the users' point of view.

In this paper we illustrate the utility of this approach for understanding the needs of CAD users. In the context of a relatively short-term observation, a combination of informal methods and formal recordings enabled us to develop a better understanding of user needs.

Site Visit - Background

The users in this study are architects at a US Army Corps of Engineers District office. They are members of an architecture department consisting of 11 registered architects, 3 draftsmen (called "techs"), and a manager, who is also an architect. The architects perform design and drafting tasks involving decision-making, whereas the techs mainly make changes to drawings constructed by the architects. The office designs government facilities all over the world and use Intergraph's MicroStation (a sophisticated CAD system) to design these buildings. It also provides other engineering services related to building design such as civil and structural engineering, all of which use the same CAD system to produce drawings.

The site visit lasted two-weeks. During that period, the first author observed users of the CAD system and engaged them in open-ended discussions. The results were recorded in field notes. In addition, some of the computer sessions were video-taped, and the users' keystrokes were recorded.

Seven architects and three techs were observed while they interacted with the CAD system performing design and drawing tasks. Of these, five architects and two techs were working on a large building for the Department of Defense. The observations were followed by open-ended discussions. Although the observer did not have any previous experience in using ethnographic techniques, he had been recently introduced to ethnography through a tutorial with Forsythe. In addition, his background in both architecture and MicroStation along with an "official" recommendation from CERL helped in building a rapport with the users.

Site Visit - Results

Based on observations made during the first week, the observer decided to focus in the second week on differences between users in window and menu usage, command vocabulary, social interaction, and resource utilization.

Differences Between Users

The techs and architects both seemed to be at the same level of proficiency. They all used a small set of primitive commands that did not vary much between users or tasks. The only obvious differences occurred in speed and window use. The techs tended to execute extremely repetitive tasks with few social interruptions and used commands to complete tasks at a much higher speed than the architects. The architects, on the other hand, tended to work on design tasks requiring problem solving, had many social interruptions, and interacted with commands at a much slower rate.

There were also differences in window usage among the architects. Some architects used four windows on one screen, some used one window on each of two screens, and some used only one screen and an occasional window. Setting-up the windows was simple, and the setup could be saved for the next session. The only problem occurred when another user needed access to the same design file and changed the settings. In contrast to the architects, all three techs used a single screen with one window. (The problem of multiple users changing a window setup could be handled by a minor design change by the software vendor. For example, the window setup could be stored in the user's login instead of the design file. This would enable all design files that are accessed by the same user to have a constant window setup).

While observing the techs in their highly repetitive tasks (using identical sequences of commands repeatedly), the observer noticed that some of the commands were buried deep in the menu hierarchies. Accessing these commands during these repetitive tasks appeared to slow down the user. It was therefore hypothesized that this could be an area in which an adaptive system might be useful. The system might detect these repeated sequences and bring up the commands in a 'sequence palette'. A double click on the active command could activate the next command in the sequence. The user could break the sequence anytime by selecting any other command in the system. This idea received much interest by the techs, but not by the architects. This was understandable as the architects typically were not involved in repetitious tasks.

Command Vocabulary

The following incident occurred during the observations:

Example 1.

One of the architects, whose project deadline was nine days away, was plotting several design files. He opened a design file by keying-in the name of the CAD application and the design file name (for example 'ustation designfile1') at the operating system level. After he had entered the application and plotted the file, he exited the entire application by pulling down the 'File' menu and selecting 'Exit'. He repeated this operation seven times, each time waiting several minutes to bring up the application and another several minutes to display the design file. The observer asked him if it was possible to open a new design file without exiting the current design file. He answered approximately "that might be possible, but this works".

This incident brings up several interesting points.

1. The user did not use the 'File Open' command because he did not know of its existence.

2. Even though the 'File Open' command was displayed on the same pull-down menu that he was using to exit, he had not noticed it. (The 'File Open' command was in fact the second entry in the menu and the 'Exit' command was the last)

3. The user seemed reluctant to try another way to do what he was doing.

In subsequent observations, it was noticed that six other users exited the application to enter a new design file. The users, keying-in the design file names, had to remember these names, which were quite long, and committed frequent typing errors.

From these and similar observations, a general pattern of sub-optimal use emerged. The users had leveled-off or 'plateaued' in their use of the CAD system and were not learning new commands. This was unexpected as the users had tremendous resources at their disposal. They were using one of the best CAD systems in the industry (offering more than 2000 commands), had formal training in its use, had up-to-date documents, had a help facility, had system support from Intergraph (the CAD vendor) and access to expert CAD users in other departments in the building. Despite these resources, the users were using only a small set of primitive commands to design and detail a fifty million dollar building.

Unused Resources

These observations raised the question: 'Given the resources, what is the reason for the sub-optimal use of the CAD system?'

Answers to this question were sought through informal interviews and revealed problems with each of the available resources.

1. Training was perceived as a once-in-a-lifetime requirement. Most users had completed their formal training more than five years ago. In the meantime, the CAD system had undergone major revisions leading to an ever-growing repertoire of commands. The users had no knowledge of these new commands and did not know about README files containing information of the new features.

2. None of the users interviewed ever remembered using documents or help. When one of them was asked to demonstrate the use of the 'Help' command, he at first had trouble finding it (even though it was very close to his cursor) and then had no idea how to use it.

3. The architects had no direct contact with the CAD vendor for any application support. There was an internal rule that all questions had to go to the system coordinator supervising all the departments in the section. The system coordinator would then pose the question to the CAD vendor and transmit the answer back to the person who asked the question. Communication problems occurred as the system coordinator spoke in computer terms and the architects spoke in command usage and architectural (that is, application) terms. This line of communication had been used unsuccessfully by one architect, who had the role of CAD coordinator within the group.

4. When the users were asked how they had last learned a new command, all of them stated that they had done so informally through conversations with other users or chance observations. Subsequently, it was observed that users exchanged information based on spatial proximity and friendship. For example, a group of three architects and a tech, who were spatially separated from the rest of the department, knew some advanced commands that the rest did not know and vice versa. Contact with outside groups (structural and civil engineering) was minimal.

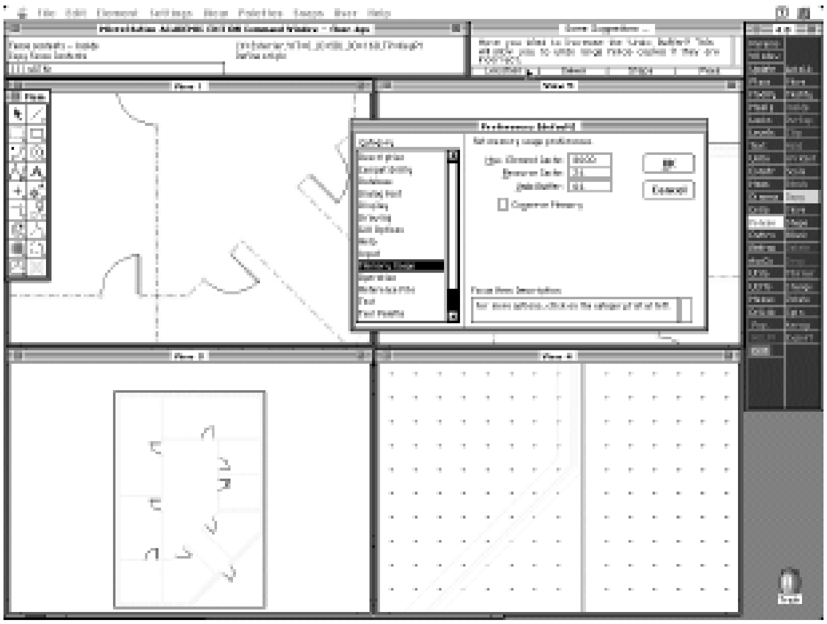

Figure 2. MicroStation interface with the Active Assistance window providing advice.

The overall tendency was to use the system to 'get the job done'. The users had no way of knowing that there were better ways of executing a task. The help and document resources were voluminous, yet passive. Informal contacts to learn new commands were minimal, and there was little input of new ideas from within the facility or from the CAD vendor.

The problem of users not being aware of powerful commands that are easily accessed through the interface is illustrated in Example 2 below. The context is a meeting of all the users in the architectural department, where ten different commands are being demonstrated by an architect (A1) and tech. Both of these users had been trained by the observer in the use of the new commands and had agreed to demonstrate them to the rest of the group. Eight of the ten commands had never been used by anyone in the group prior to the study. The other two were being used only by the group that was spatially separated from the rest of the department.

Example 2.

A1: "How many people know what this arrow..."

A2: "The arrow?"

A1: "The arrow up here....in the main menu". (A1 is pointing to an icon in the shape of an arrow. It is in a very prominent position on the top left-hand corner of the main menu).

A2: "I haven't used it."

Manager: (in jest) "Raise your hands if you know it. A3?" (A3 has the most complaints about using the system).

(Laughter).

Example 3 below illustrates users recognizing the power of an unused command as well as surprise by the manager that it was not being used. The discussion occurred after the demonstrations had been completed and most participants had left the area.

Example 3.

A4: "With these new commands we ought to a.....save..a...tenth of the time."

A1: "Like the 'Ctrl-O', when you plot drawings". ('Ctrl-O' is the key board short cut to using the 'File Open' command. The key board short cut is mentioned alongside the 'File Open' command on the menu)."

A5: "Save time."

A1: "Save a lot of time."

Manager: (Whistles) "Just can't believe you guys didn't know this.....I mean...." (laughs self consciously).

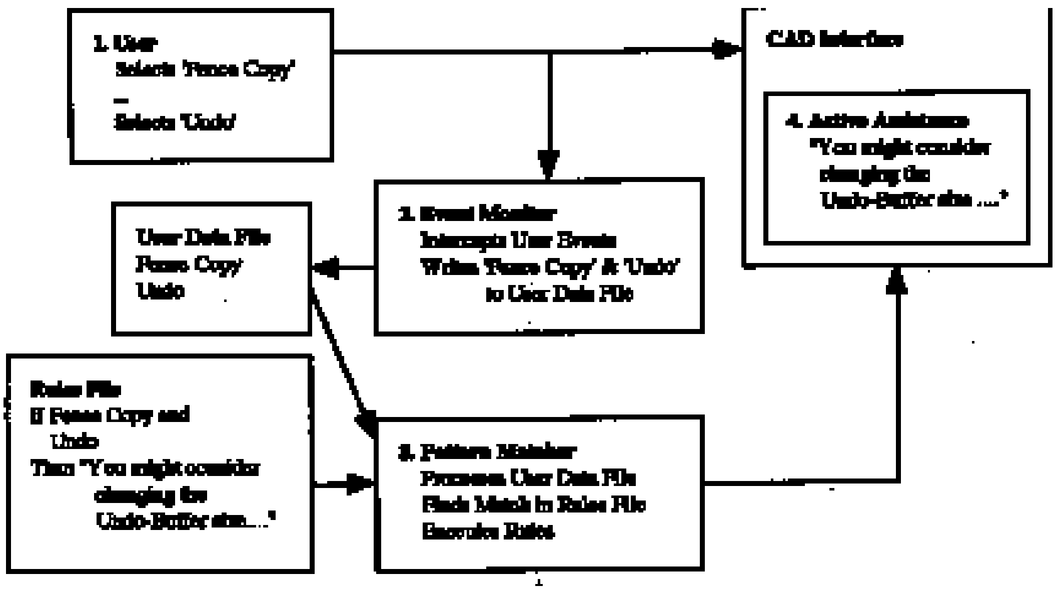

Figure 3. Architecture of the Active Assistant System.

As these examples illustrate, the observer in this study moved over time from participant observation to active intervention. In the latter role, he pointed out to the users important commands that they did not use. He did so for two major reasons: 1. to gain initial insights into the motivation of and interactions between the users under observation and 2. to provide an immediate 'pay-back' to them for the time they spent cooperating in the study.

Emergent Hypothesis

The above discussions and observations showed that the users had leveled-off in learning the system and were using it in a sub-optimal way. They seemed more interested in simply 'getting the job done' than in learning new and better ways to use the system. Not being aware of better ways, they did not even know how to ask questions or explore alternative uses.

Although productivity growth has many aspects that include CAD management and cooperative behavior (Gantt & Nardi, 1992), we limit our discussion here to appropriate technologies that address the observed problems directly. During the site visit, the observer hypothesized that users might benefit from an active assistant system that constantly monitors their command usage and presents better ways of doing the same tasks. (The concept of active assistance will be explored in the next section). However, the issue of motivation was troubling. For instance, in Example 1 above, the architect was not motivated to explore a new command. How useful could an active assistant system be if it suggested a new command to a user who was not motivated to explore it? And why was it that the command was later recognized as a powerful command?

Some hints emerged in follow-up discussions. When asked, a tech said that he would feel motivated to use a suggestion provided by the system if it had been used by another user in the group. In this way, he would know that the command was useful, that it worked, and that he could talk to that person if he had trouble using the command. He also said that he would not follow up on any suggestion if he was pushing a deadline.

Based on these discussions, the observer concluded the field trip with a new and refined hypothesis: The users need an active assistant system that constantly monitors their command usage and presents better ways of doing the same tasks. The suggestions must be presented in a way that motivates the user to explore the suggestions and must be sensitive to issues like timing and form of presentation.

A Prototype for Active CAD Assistance

We are currently in the process of testing the hypothesis of active assistance through a prototype. This prototype monitors command usage and provides suggestions to improve it. It is designed using the MicroStation Development Language (MDL) and can be loaded into the MicroStation environment. When loaded, the system places a dialog box (entitled "Some Suggestions") in the upper right hand corner of the screen as shown in Figure 2.

As the user interacts with the CAD environment, messages appear in the dialog box depending on the user's interaction. These messages typically provide suggestions about command usage that might be useful in the current context. The dialog box also contains a button named 'Location'. Selecting the 'Location' button opens appropriate dialog boxes or palettes that show the location of a command when one has been suggested. We are exploring additional buttons that will provide quick demos as well as detailed steps to explain the use of the commands being suggested.

The above approach is best illustrated by a common mistake in using the 'Fence Copy' command. This is a MicroStation command that allows the user to group many elements within a polygon called a 'fence' and copy them to another location. Users reported that a large fence may contain a large number of elements and exceed the undo-buffer size. If the user makes a mistake in selecting the fence contents or the location where it is to be copied, the fence copy operation cannot be completely undone, and the user has to undo the mistake manually. This can be tedious and time-consuming. MicroStation provides the capability of changing the undo-buffer size by keying-in a new value in a parameter field. However, the users in this study were unaware of this capability and reported making frequent manual changes to rectify incorrect fence copies.

The 'Fence Copy' example is one of the situations that we have incorporated into the active assistant prototype. If the user selects the 'Fence Copy' command followed by the 'Undo' command, a message appears in the dialog box. The message suggests that the user change the size of the undo-buffer if the content of the fence is large. Subsequently, if the 'Location' button on the dialog box is selected, the system responds by opening the dialog box where the undo-buffer field can be changed. This demonstrates how the active assistant system treats an error committed by the user as an opportunity for the user to learn more about the system.

It can be argued that this problem occurs as a result of bad interface design. For example, the problem could be alleviated if the 'Fence Copy' command itself disallows the operation or warns the user when the undo-buffer size has been exceeded and, possibly, shows how to change it. However, this requires that the vendor is convinced of the value of such an interface redesign and is willing to implement it, which is outside the scope of this study. In either case, the main issues remain: the error must be known, it must be detected, and the user must be notified about ways to avoid it.

Figure 3 shows the basic architecture of the active assistant system under development. It consists of four components; an event monitor, a user data file, a pattern matcher, and a rules file. When a user selects a command (1), the event monitor writes that event to the data file (2). The pattern checker processes sets of these events to find a match with any rules in the rules file (3). If the pattern of events matches the antecedent of a rule, this rule is executed, and suggestions from the consequent of the rule appear in the active assistant dialog box (4).

A complementary part of our current research focuses on identifying the nature and causes of CAD interaction errors like the one described above. This is being done by applying formal techniques of analysis and modeling such as NGOMSL (Kieras et al., 1994) to the data from the field study. We expect that a deeper understanding of the nature of CAD errors and their underlying causes will enable us to design better CAD interfaces and to provide active assistance where appropriate.

It is unclear at the present time how motivation factors into the design of active assistance. The issues of attention and intervention have plagued other attempts at developing active systems. For example, usability studies on intelligent help systems (Carroll and Aaronson, 1988) and design critiques (Fisher et al., 1991) have shown that users frequently ignore, do not notice, or misunderstand computer generated advice. Finding answers to such issues requires further on-site observations once the prototype system has been completed and deployed in the real world.

Conclusion

Faced with the productivity puzzle, we concluded that it was more important to understand CAD users and their tasks before attempting to develop more technology. Inspired by the success of studies in related fields, we explored the possibility of using ethnographic techniques developed by cultural anthropologists to study CAD users in their real-world environment.

During the study, we realized that our users had leveled-off at a sub-optimal use of the system. Furthermore, none of the information sources available to them helped them explore new functionality because they were passive. The users did not know that there existed better ways to execute a task and therefore did not know when to ask for them. They also were not motivated to explore the system because they were mostly in the mode of 'getting the job done'. We therefore arrived at a hypothesis: users required an active assistance system that made suggestions while they were 'getting the job done'. The prototype of such a system is under development and will be tested in further site visits with end-users. Questions about the significance of motivation remain and require further study.

We conclude that ethnography helped us, in a period of two weeks, to formulate a hypothesis based in reality. It also made us aware of usability issues that should be considered early in the design of a CAD system and its interface. Although one field trip is not enough to validate a hypothesis, it points to seemingly promising directions for research and future visits. The relatively inexpensive nature of these visits makes this approach an attractive alternative to testing during or after a project has commenced which may lead to possibly costly mistakes that are harder to correct.

ACKNOWLEDGMENTS

This research was supported by the United States Army Corps of Engineers Construction Engineering Research Laboratory (USA-CERL). We thank the users in this study for their time and cooperation. We also thank Bentley Systems for the academic edition of MicroStation.

Bietz, W., Langner, T., Luczak, H., Muller, T. and Springer, J., "Evaluation of a compact CAD-course in laboratory experiments", International Journal of Human Computer Interaction, (1990): vol. 2, no. 2, 111-35.
Carroll, J. M. and Aaronson, A. P., "Learning by doing with simulated help", Communications of the ACM, (1988): vol. 31, no. 9, 1064-1079.
Fisher, G., Lemke, A. C., Mastaglio, T. and Morch A. I., "The Role of Critiquing in Cooperative Problem Solving", ACM Transactions on Information Systems, (1991): vol. 9, no. 3, 123-151.
Forester, T., Computers in the Human Context (New York: Basil Blackwell, 1989
Forsythe, D. E., "Blaming the User in Medical Informatics: The Cultural Nature of Scientific Practice", Knowledge and Society , (1992): vol. 9, 95-111.
Forsythe, D. E., "Using Ethnography in the Design of an Explanation System" Expert Systems With Applications, (in press).
Gantt, M. and Nardi, B., "Gardeners and gurus: Patterns of cooperation among CAD users", Proceedings of CHI' 92, (1992): 107-117.
Kieras, D., "A Guide to GOMS Task Analysis", Unpublished manuscript, University of Michigan, (1994a).
Luczak, H., Beitz, W., Springer, J. and Langner, T., "Frictions and frustrations in creative-informatory work with computer aided design-CAD-systems", Proceedings of the Fourth International Conference on Human-Computer Interaction, (1991): 175-9.
Lundsgaarde, H. P., "Evaluating Medical Expert Systems", Social Science and Medicine, (1987): 24(10):805-819.
Majchrzak, A., "Effect of CAD on the jobs of drafters and engineers: a quantitative case study", International Journal of Man-Machine Studies, (1990): vol. 32 no. 3, 245-62.
Nardi, B., A Small Matter of Programming: Perspectives on End-User Computing (Cambridge, MA: MIT Press, 1993).
Powdermaker, H., Stranger and Friend: The Way of an Anthropologist (New York: W. W. Norton & Company Inc., 1966).
PSMJ, CADD application and user survey (Professional Services Management Journal, 1994).
Suchman, L. A., Plans and Situated Actions (Cambridge, England: Cambridge University Press, 1987).
Werner, O. and Schoepfle, G., Systematic Fieldwork (Newbury Park, CA: Sage, 1987).