ELSEVIER

Automation in Construction 5 (1996) 243–255

AUTOMATION IN CONSTRUCTION

CAD usage in an architectural office: from observations to active assistance [1]

Suresh K. Bhavnani [a,*], Ulrich Flemming [a], Diana E. Forsythe [b,2], James H. Garrett [c], Doris S. Shaw [d], Albert Tsai [e]

[a] *Department of Architecture and Engineering Design Research Center (EDRC) Carnegie Mellon University, Pittsburgh, PA 15213, USA*
[b] *Intelligent Systems Laboratory, Department of Computer Science, University of Pittsburgh, Pittsburgh, PA 15260, USA*
[c] *Department of Civil and Environmental Engineering and Engineering Design Research Center (EDRC) Carnegie Mellon University, Pittsburgh, PA 15213, USA*
[d] *Construction Engineering Research Laboratory (CERL), US Army Corps of Engineers, Champaign, IL 61826, USA*
[e] *University of Illinois, Urbana—Champaign, Department of Computer Science, Room 2413, 1304 West Springfield Avenue, Urbana, IL 61801, USA*

Abstract

The functionality and resources provided by CAD systems have been increasing rapidly, but 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 arrive at such an understanding, the first author visited a federal architectural office and observed CAD users in their natural setting. This paper describes preliminary results obtained from the study, which used ethnographic techniques developed by cultural anthropologists. The study revealed that users had leveled-off in their learning and experimentation and were using the CAD system in suboptimal ways. By asking why users were not using many resources available to them to improve performance, the observer uncovered issues of communication and management that needed to be addressed. Based on this understanding, the authors provide explicit recommendations to CAD users and vendors. In addition, they hypothesize that users might benefit from a system that provides active assistance, that is, intervenes spontaneously with advice, assistance, and relevant information while the user interacts with the CAD system. They conclude with some issues revealed by the study that should be considered when developing such active assistance.

Keywords: CAD usage; Participant observation; Active assistence

1. Introduction

Productivity increases through the use of computers have been negligible or difficult to achieve in various application domains [1]. The huge investments in the computer revolution, in general, have apparently not paid off in terms of productivity growth [2], a phenomenon that is commonly referred to as the *productivity puzzle*. Productivity in firms

* Corresponding author. Fax: +1-412-2686129. E-mail: suresh@andrew.cmu.edu.

[1] Discussion is open until March 1997 (please submit your discussion paper to the Editor of Architecture, Y.E. Kalay).

[2] Present address: Division of Medical Anthropology, Department of Epidemiology and Biostatistics, University of California, 1350 Seventh Avenue, Room 302, San Francisco, CA 94143-1850, USA.

0926-5805/96/$15.00 Copyright © 1996 Elsevier Science B.V. All rights reserved.
PII S0926-5805(96)00149-5

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 (leveling off at a median value of 25%) until they have worked with CAD for five years [3].

The few laboratory and field studies on CAD usage that are available present a dismal picture indeed. Bietz et al. [4] 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. [5] 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 [6] 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 appropriately. Others blame the CAD system for having poorly designed and unnatural interfaces, non-adaptive interfaces, inadequate and unstable functionality, and poorly designed help, training, and documentation.

This is unfortunate as there is a good-faith effort on the part of CAD developers as well as users to make effective use of CAD. The first author, while being employed at a CAD company for many years, observed several departments in the company as well as customers spending considerable resources to reap the benefits of CAD. The company offered extensive training, documentation, on-line help, phone support, and organized discussion forums such as user group meetings and steering committees. New customers underwent extensive training, and phone-support personnel were regularly inundated with user calls. In addition, a growing number of books have been available to teach users how to use various CAD systems.

Why does the use of CAD systems show such disappointing results despite the extensive infrastructure built to support CAD users? The general importance of effective management, training, and communication is well-known, but how do these factors play out in specific contexts? And are there other aspects in an office environment that contribute to the productivity puzzle? CAD user groups and steering committees provide a forum for the exchange of ideas between developers and users, but may not necessarily reflect what actually happens in an office. After all, the users chosen from a company to attend these meetings tend to be the most vocal and computer-literate ones and may not represent the normal user. Is it possible that usage is affected by unexpressed values and hidden assumptions? Finally, can we use newer technologies at our disposal to improve usage (see [7] for a discussion of adaptive CAD interfaces)?

These questions motivated us to observe real users doing CAD work in their natural work environment. We hoped that an in-depth case study of end-users would provide us with first-hand observations of successes and failures in CAD usage. This paper describes preliminary findings from this study and offers some suggestions to users, managers, and vendors, as well as a hypothesis for future system design.

2. 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 [8,9]. Their method is known as *ethnography*. Originally applied in non-Western societies, ethnographic techniques have been used more recently to illuminate such problems as computer usability [10–13]. For example, Forsythe demonstrated how ethnography can be used to help design a computer-based explanation system for migraine sufferers [14]. The study included extensive observation of interactions between doctors and patients in clinical settings as well as in-depth interviews with migraine sufferers.

The main ethnographic data-gathering method, known as *participant observation*, involves the use of unobtrusive observational techniques. While building rapport with users, trained fieldworkers systematically immerse themselves in the users' work environment. This 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 an understanding aids the fieldworker in interpreting observational data. Participant observation may be supplemented — as appropriate — by interviews, 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 arriving at an understanding of CAD usage in architectural offices. In the context of a relatively short-term observation, a combination of informal methods and formal recordings enabled us to gain insights into the issues raised and arrive at various recommendation to CAD users, managers, system developers and vendors.

3. Site visit — background

The users in this study are architects at a US Army Corps of Engineers District office. They are members of the architectural section of the design branch 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 uses MicroStation (a sophisticated CAD system developed by Bentley Systems, Inc.) 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 the Corps of Engineers Construction Engineering Research Laboratories (CERL), the sponsor of this study, helped in building rapport with users.

4. 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.

4.1. Differences between users

The techs and architects 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.

Architects also differed among themselves in window usage. 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 (see Fig. 1). 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 tasks appeared to slow the user down. 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. The observer also demonstrated the use of the palette builder command that

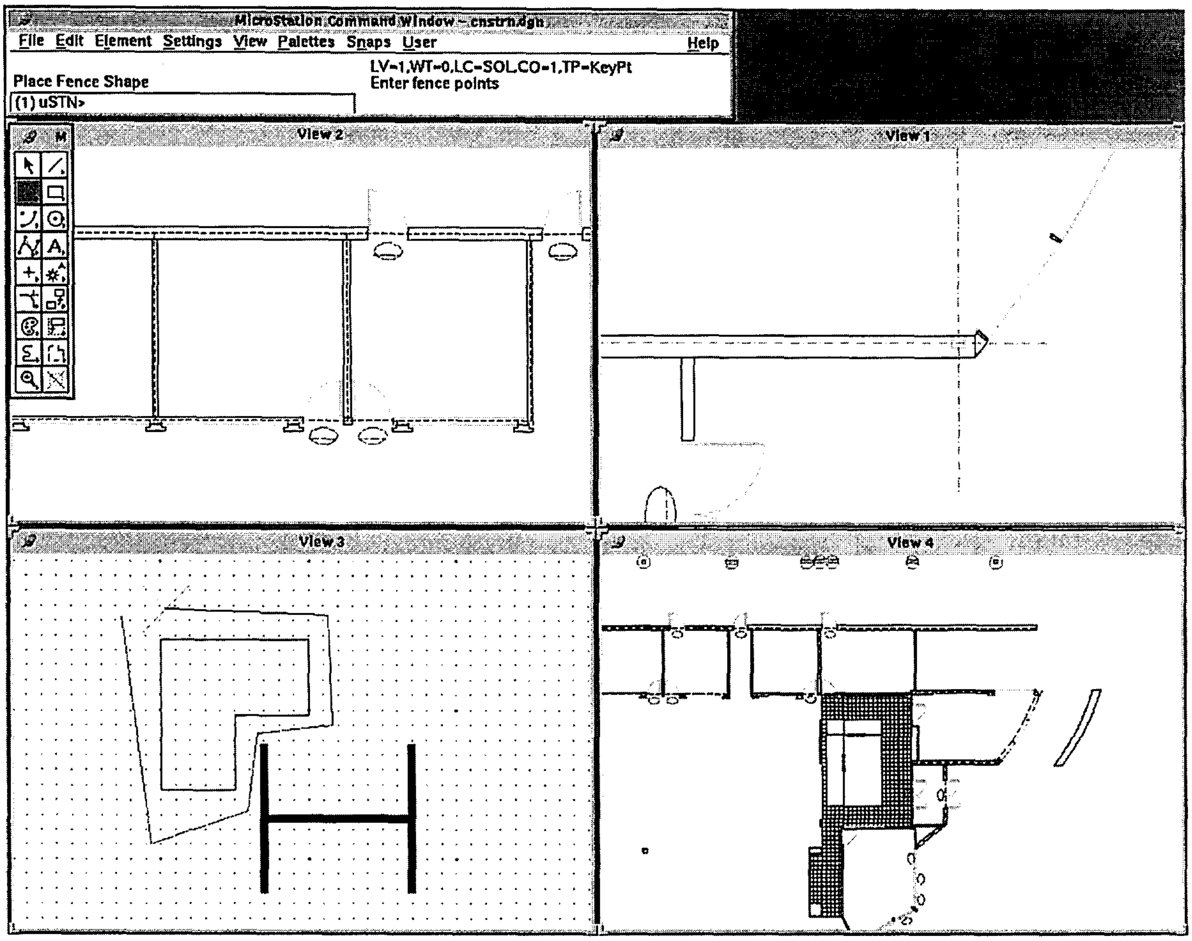

Fig. 1. MicroStation interface showing typical window and menu set-up.

could allow a tech to set up his own palette rapidly for a task.

4.2. 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 suboptimal 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 received 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 sections 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.

4.3. Unused resources

These observations raised the question: Given the resources, what is the reason for the suboptimal 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 event. All the users had completed their formal training more than two and a half years ago. In the meantime, the CAD system had undergone major revisions leading to an ever-growing repertoire of commands. The users had little 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 sections in the branch. From discussions with management, the observer learned that the rule was put into place to prevent the vendor's phone support from being inundated with user requests. However, this indirect communication did not work well for the architects. An architect had to pose a question first to the system coordinator, who would then pass the question to the CAD vendor and transmit the answer back to the architect 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 domain) terms. This line of communication had been used unsuc-

cessfully 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 architectural section, knew some advanced commands that the rest did not know and vice versa. Contact with outside groups (structural and civil engineers) was minimal.

The overall attitude was 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 section, 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 section.

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. 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 keyboard short-cut to using the FILE OPEN command. This 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).

The above examples demonstrate how the method of participant-observation reveals important aspects of a social situation. The manager, who does not use the CAD system to create drawings, expresses surprise that his group does not know important commands in the CAD system. Such observations reveal that the manager assumed his architects were using the CAD system efficiently. This is in sharp contrast to the view shared by architects who regularly use the CAD system as demonstrated in the following examples. Note also the importance of informal utterances such as whistling or laughter that are important in ethnographic studies but may not be treated as data in other methods.

Example 4.

A6: "There are a lot of things that I don't... know exist on this as of yet, it's still... probably doing some things wrong, or not as fast as I possibly could because... just not aware of a better way of doing it sometimes."

Example 5.

A7: "... MicroStation is so extensive... I mean there are so many things that I never even use uh... and it takes quite a bit of time to learn how to use it."

These examples demonstrate that the architects realize they are not making optimal use of the system. The discussion therefore helped identify the manager's misconception of CAD usage in his section, which is possibly related to the view that training is a once-in-a-lifetime requirement. While such tacit assumptions are important in finding ways to improve the effective use of CAD, they are less likely to emerge in decontextualized laboratory experiments or even in structured interviews and questionnaires, where the questions are often constrained by the researcher's preconceived notions.

Examples 2 and 3 also illustrate how 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: (i) to gain additional insights into the motivation of and interactions between the users under observation, and (ii) to provide an immediate pay-back to them for the time they spent cooperating in the study. The users have welcomed the idea of future site visits, and the observer has had continued interactions with them over the phone.

5. Recommendations

As described in this paper, participant observation can yield a wide range of data leading to a richer understanding of a social situation. This new understanding can suggest different courses of action. Our experience at the federal office led to the following specific recommendations to users, managers, and vendors:

1. The repetitive nature of the tasks executed by the techs suggests that techs could create a customized palette relevant to completing a repetitive task. We therefore demonstrated and recommended the use of the palette builder command (PALBLD) provided by MicroStation to create quickly a customized palette. By using this customized palette, the techs could avoid having to go repeatedly through deep menus. In follow-up calls to the office, we have learned that one tech has begun to use this feature to create his customized palette.
2. The internal management rule that prevented architects from directly accessing the vendor phone support has led to many lost opportunities for improving usage. As management was concerned that too many people would use the phone support, we recommended that the CAD coordinator in the architectural section be given the authority to make calls that were necessary for the group. This, we hoped, would more closely reflect the needs of the architectural group. The recommendation was well received by the management as well as by the CAD coordinator.
3. Discussions on improving CAD usage occurred only during chance encounters, as when one architect looked over the shoulder of another or when two architects were sitting close to one another. Therefore, we recommended that the CAD coordinator's role be extended to include regularly observing the usage of his peers. This observation might help reveal inefficiencies and encourage discussions about CAD usage. (See [15] for a discussion of the usefulness of local experts in a CAD organization).
4. We have communicated to the CAD vendor the usefulness of conducting on-site observational studies to understand the realities of using CAD as demonstrated in this study. For example, to better reflect and encourage the way users exchange and learn information in an office environment, vendor training could be redesigned to provide group exercises, where users from an organization have to interact and share information in order to complete a project.
5. As there was limited contact between sections and therefore little sharing of CAD experience within the federal office, we recommended that there be regular meetings with other sections to discuss CAD usage. We also recommended more frequent training based on major upgrades in the software provided by the vendor.
6. We are currently analyzing the video tape and keystroke data of the CAD interactions collected during the site visit. This keystroke level analysis has begun to reveal many inefficient strategies used to complete drawing tasks [16]. The analyses appear to suggest that, in addition to learning how to use CAD commands, users also require training on how to decompose a drawing task to make effective use of the CAD medium.
7. As our sponsors at CERL have been involved in the research and development of computer-aided instruction (CAI), we were motivated to see if there were any opportunities for new computer aids that could help improve CAD usage. The following section describes the design of such a system.

6. A hypothesis for active CAD assistance

As described earlier, the observed users concentrated on getting the job done and spent little time to learn new and better ways to use the system. The observer therefore hypothesized that users might benefit from an *active* assistant system that monitors their command usage and presents better ways of doing the same tasks at appropriate times. During the second week, after the users had been video-taped performing CAD tasks, the observer brought up the concept of active assistance in order to get some immediate feedback. The following is a typical response to this suggestion.

Example 6.

A7: "... you get to a point on the CAD... it's not that you don't want to learn anymore, but you're comfortable with your level of use and you can do your job... and you can do your job adequately let's say and... and... you're not really experimenting with it as much anymore to see what you can do... whereas if you have some kind of a... tutorial I think... uh then it kind a continues you... your education... of course its still up to the user... it's all up to the user... what he wants to do... whether or not he wants to use the command or..."

In discussions about the design and workings of an active assistance system, this user declared that he would like a tutorial window that appears in the upper right-hand corner next to the MicroStation command window in an area that was unused. He would like to be able to 'blow it away' if he did not need it. He also indicated that he would like the system itself to open the menu and highlight the command being suggested so he would not have to search for it. While referring to the help window (that he had brought up for purposes of demonstration, but had never used before), he described how active assistance might work:

Example 7.

A7: "OK let's say up here it tells you... YOU COULD HAVE DONE THAT COMMAND BY... ah... COPY AND INCREMENT TEXT as we have up here (points to text in the help window with the cursor) WOULD YOU LIKE TO SEE MORE? and I could toggle YES and when I did YES it would take like View 1 and it would show me in View 1 how the command is... how to do the command, it would just do it... it'll do it. STEP 1 — it would show me PLACE DATA POINT or whatever."

During this discussion, the user described in detail how graphics would help explain the workings of a command more rapidly than just textual explanation as provided by the existing help window.

While these design ideas were instructive, 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 other discussions. When asked, a tech said that he would feel motivated to use a command 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 stated that he would not follow up on any suggestion if he was pushing a deadline. In such situations, he would prefer the system to store the suggestions in a file that he could print out and read at a later time.

Based on these discussions, the observer concluded that the active assistant concept appeared promising, although its success would depend on its ability to motivate a user to attend, comprehend, and use the suggestions. Such issues need further investigations through the development of prototypes and further user studies.

7. A prototype for active CAD assistance

We have developed the prototype of an active assistance system based on discussions with users as described above as well as on our ongoing analysis of the video and keystroke data collected. In the data, one of the architects used individual lines to draw an orthogonal shape. When he chose to mirror and copy the shape, he used a fence shape to group the lines followed by the FENCE MIRROR–COPY command. Subsequently, when he chose to pattern the two shapes he had just drawn, he manually patterned it with individual dots and triangles representing concrete by copying them from an adjacent shape. The creation of a fence and the manual patterning of a shape required many precise cursor inputs, thus leading to many motor slips (errors caused inadvertently by inaccurate mouse selections) and a marked increase in the total number of mouse and keystroke interactions needed to complete the task (see [16] for a detailed discussion). Drawing methods requiring accurate inputs could have been avoided if the user had used the orthogonal shape command to draw the shape. This would allow the shapes to be patterned automatically using the pattern command and to be mirror-copied directly without using a fence.

Using the above example to test our hypothesis, we have implemented the Active Assistant (AA), a system that detects when a user attempts to perform any fence manipulation on a shape drawn with individual lines. When this occurs, AA suggests a more efficient way to complete the task. We first describe how a user might interact with AA and then describe its architecture, which is able to capture also other kinds of situations and offer remediation.

7.1. Interacting with the Active Assistant

The AA interface was designed to provide active assistance at the following two levels:

1. Textual notification that there is a better way to perform a task executed by the user.
2. Graphic remediation describing what was done and how to improve it.

When AA is invoked, a window named ACTIVE

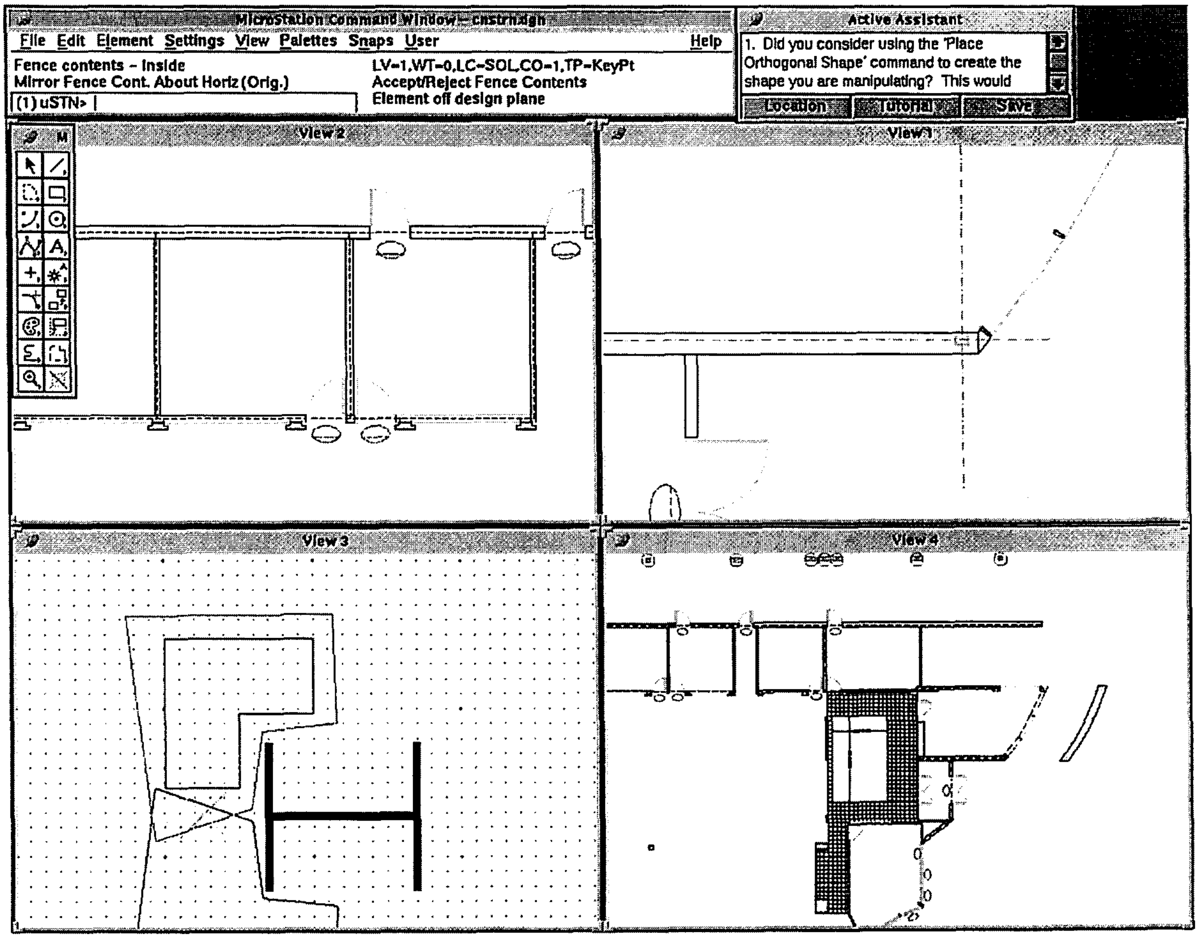

Fig. 2. The Active Assistant window with textual suggestions.

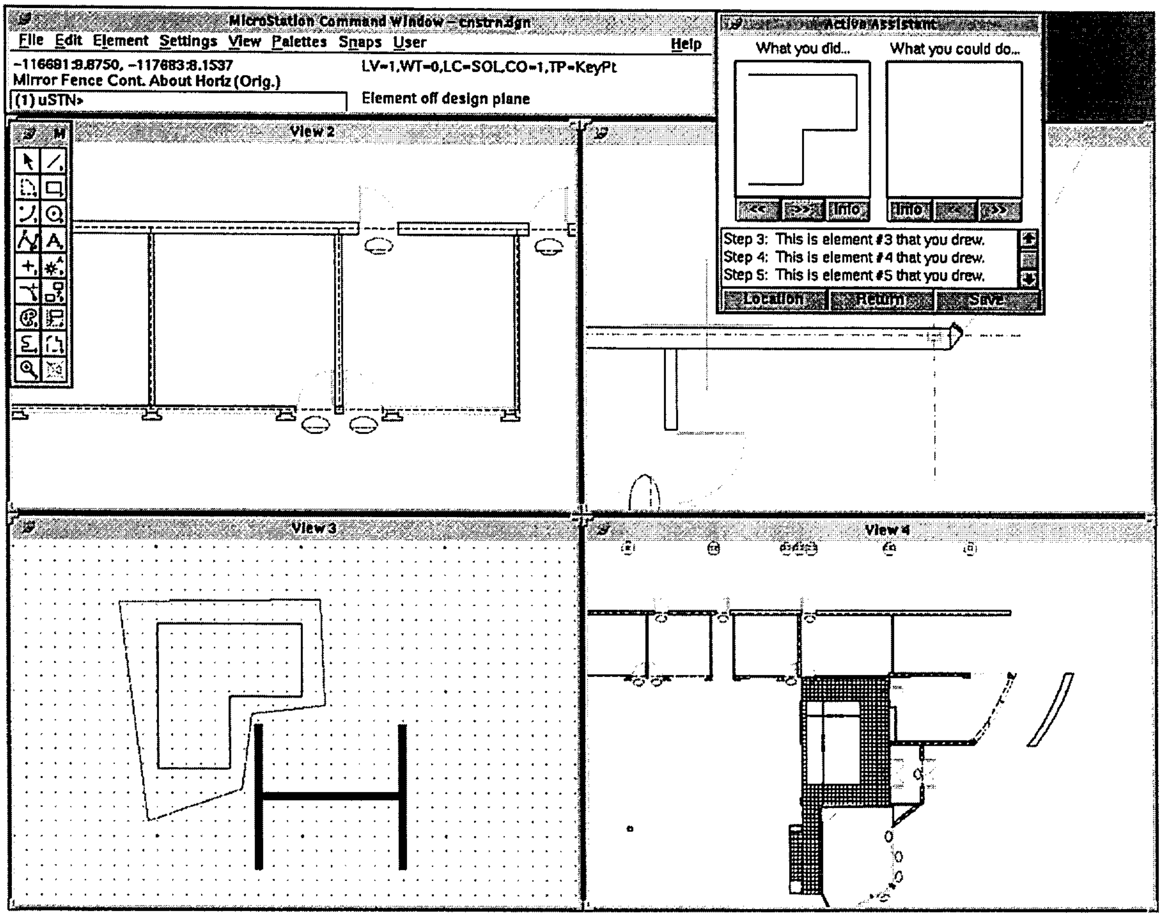

Fig. 3. The tutorial window enables users to inspect the steps they used to perform a task.

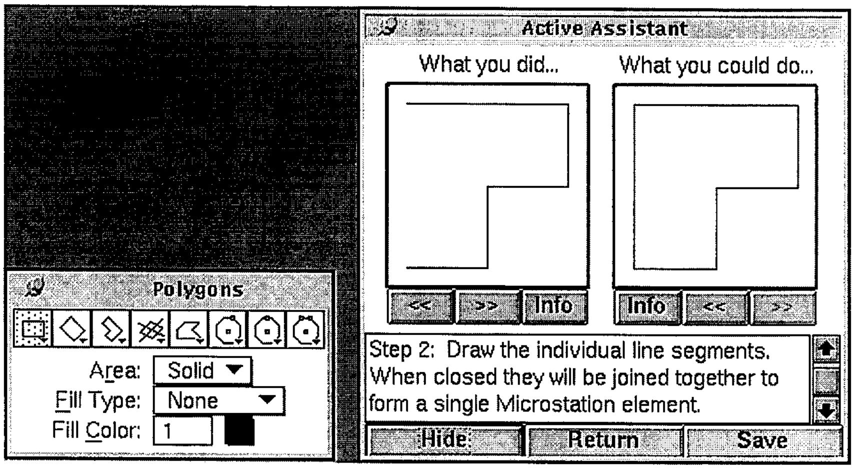

Fig. 4. The tutorial window enables users to inspect steps they could use to perform their task more efficiently. When the LOCATION button is selected, it opens the appropriate menu where the command being suggested resides; HIDE reverses that action.

ASSISTANT is displayed at the top right-hand side of the screen. As shown in the lower left-hand view of Fig. 2, if a user draws a closed shape with primitive lines and then attempts to perform a fence operation on them, (for example, FENCE–MOVE or FENCE–COPY), the system provides textual feedback in the Active Assistant window. In this example, it is suggesting the use of the PLACE ORTHOGONAL SHAPE command.

If the user selects the LOCATION button, the system opens the appropriate menu where the command being suggested resides. If the user selects the SAVE button, the text in the scrollable field is written to a file that can be printed for later reference.

If the user selects the TUTORIAL button, the window enlarges as shown in Fig. 3. This tutorial window contains two areas on the top named WHAT YOU DID and WHAT YOU COULD DO. If the user selects the forward and backward arrow buttons under the WHAT YOU DID box, the system graphically recreates the steps the user performed to draw the lines in the design file. The text box at the bottom of the window provides a description for each of the steps. The INFO button lists the disadvantages of the method selected by the user to draw the shape. In this example, it would encourage the user to try and move the shape by selecting one of its edges, in which case only the selected line will move.

The forward and backward arrow buttons under the WHAT YOU COULD DO box can be used to display graphically the steps that could be used to draw the same shape in a more efficient way as shown in Fig. 4. The text field below the graphic box provides a textual description of each step needed to perform the task more efficiently. The INFO button provides the advantages for the method recommended. In this example, the user is encouraged to move the shape by selecting one of its edges; this time, the entire shape will move.

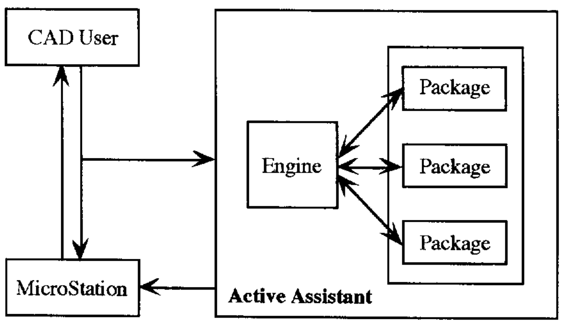

Fig. 5. Architecture of the Active Assistant.

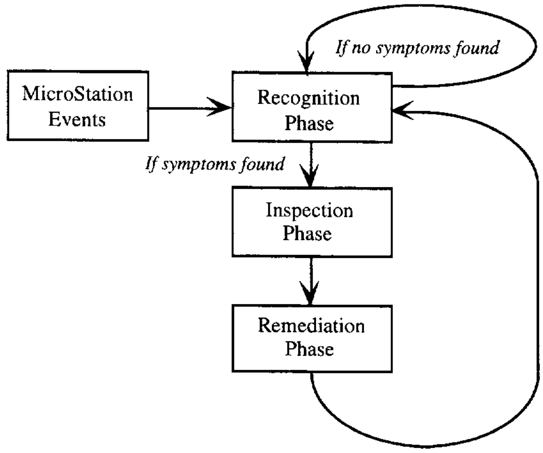

Fig. 6. Phases of operation in the Active Assistant.

As shown in Fig. 4, when the LOCATION button is selected, the appropriate menu is displayed and the suggested command highlighted. If the RETURN button is selected, the window returns to its original state as shown in Fig. 2.

7.2. Architecture of the Active Assistant

The architecture of the Active Assistant is similar to other knowledge-based systems that separate the inferencing mechanism from the domain knowledge. This separation facilitates the addition and modification of the domain knowledge. As shown in Fig. 5, AA consists of an engine and a collection of package modules containing the domain knowledge. The AA engine controls the overall operation of the system by monitoring user events and accessing appropriate packages.

Fig. 6 shows the three distinct phases in the Active Assistant's operation: recognition, inspection, and remediation. During the recognition phase, AA monitors the user for symptoms of suboptimal and incorrect CAD usage. When a symptom is recognized, AA proceeds to the inspection phase, where it collects the information necessary to create advice tailored to the user. This advice is presented to the user during the remediation phase. The transition between phases is controlled by the AA engine. The knowledge required at each stage is contained in the package modules.

As the package modules are independent, the knowledge they contain can be stored in representations appropriate to the respective domain knowledge. In our prototype, we need to determine what kind of shape has been drawn before a command can be suggested (for example, the command to draw an irregular polygon is different from the command to draw an orthogonal rotated polygon). We therefore use a decision tree to represent the tree-like structure of inspections to be made on a set of lines defining a shape. Once the type of shape has been determined, the trigger component (implemented as a hybrid of IF–THEN rules and a finite element state machine) determines what suggestions to provide the user. The AA prototype has been implemented using the MicroStation Development Language (MDL), which makes the prototype independent of hardware platforms.

We are currently exploring the design of an authoring tool for the addition and modification of knowledge contained in the packages. Additionally, we are investigating the design of a user model that stores and processes past information about a user's interaction with the AA. This information could be used to tailor the advice and behavior of the AA to a specific user.

It is unclear at the present time how user motivation affects active assistance. The issues of attention and intervention have plagued other attempts at developing active systems. For example, usability studies on intelligent help systems [17] and design critiques [18] have shown that users frequently ignore, do not notice, or misunderstand computer generated advice. Finding answers to such issues requires further on-site observations of active assistance in use.

8. Conclusion

We are aware that our user sample is small and that the office setting we have studied may not be typical of CAD usage in general. The following conclusions must therefore be seen against this background. However, it is pertinent to note that our sponsor is a rather large institution maintaining design offices at various locations. Observations leading to improved CAD usage in these offices alone would be valuable to the Corps of Engineers. Additionally we have indications that our observations are valid beyond the specific type of office in which they took place. Most encouraging among these has been our experience at various conferences and seminars at which our findings and suggestions have been presented. For example, when these results were presented at the 1994 US Intergraph Graphics Users Group (IGUG) Fall Conference, response from users, trainers, managers, and representatives from Intergraph Corporation was immediate and supportive and, so far, no one has contradicted our observations.

With this in mind, we propose the following conclusions. Given the extensive infrastructure available to support CAD users, we were initially puzzled by reports of poor performance of CAD systems. We wondered if there were aspects of CAD usage in an office environment that have not been understood and addressed. 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 the observed users had leveled-off at a suboptimal use of the system. Furthermore, they did not take advantage of the many resources available to improve performance. Further investigations uncovered issues of communication and management that need to be addressed. We also noticed that the users did not seem motivated to explore the system. Rather than seeking to use the software optimally, they were mostly interested in getting the job done using familiar (if inefficient) routines. On-line help and other facilities were not used as they were too passive to be helpful. We therefore arrived at a hypothesis: users might benefit by an active assistance system that makes suggestions while they are getting the job done. The prototype of such a system is under development and will be tested in future site visits with end-users. Questions about the significance of motivation remain and require further study.

Ethnography helped us, in a period of two weeks, to develop explicit recommendations to improve CAD usage and to formulate a hypothesis based on qualitative and quantitative data. Although one field trip is not enough to understand all the complex issues that affect usability in an office environment, 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.

Acknowledgements

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 and Bentley Systems, Inc., for the academic edition of MicroStation.

References

[1] Landauer, T.K., *The Trouble with Computers*, The MIT Press, Cambridge, MA (1995).

[2] Strassmann, P.A., *The Business Value of Computers*, The Information Economics Press, New Canaan, CT (1990).

[3] PSMJ, CADD application and user survey, *Professional Services Management Journal* (1994).

[4] 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*, *2* (2) (1990) 111–135.

[5] Luczak, H., Beitz, W. Springer, J. and Langner, T., Frictions and Frustrations in Creative-Informatory Work with Computer-Aided Design Systems, *Proceedings of the Fourth International Conference on Human-Computer Interaction* (1991) 175–179.

[6] Majchrzak, A., Effect of CAD on the jobs of drafters and engineers: a quantitative case study, *International Journal of Man-Machine Studies*, *32* (3) (1990) 245–262.

[7] Bhavnani, S.K., Garrett, J.H. and Shaw, D.S., Leading Indicators of CAD Experience, *Proceedings of CAAD Futures '93*, (1993) 313–334.

[8] Powdermaker, H., *Stranger and Friend: The Way of an Anthropologist*, W.W. Norton and Company, Inc., New York, NY (1966).

[9] Werner, O. and Schoepfle, G., *Systematic Fieldwork*, Sage, Newbury Park, CA (1987).

[10] Forsythe, D.E., Blaming the User in Medical Informatics: The Cultural Nature of Scientific Practice, *Knowledge and Society*, *9* (1992) 95–111.

[11] Lundsgaarde, H.P., Evaluating Medical Expert Systems, *Social Science and Medicine*, *24* (10) (1987) 805–819.

[12] Nardi, B., *A Small Matter of Programming: Perspectives on End-User Computing*, The MIT Press, Cambridge, MA (1993).

[13] Suchman, L.A., *Plans and Situated Actions*, Cambridge University Press, Cambridge, UK (1987).

[14] Forsythe, D.E., Using Ethnography in the Design of an Explanation System, *Expert Systems With Applications*, *8* (4) (1995) 403–417.

[15] Gantt, M. and Nardi, B., Gardeners and Gurus: Patterns of Cooperation Among CAD users, *Proceedings of CHI' 92* (1992) 107–117.

[16] Bhavnani, S.K. and John, B.E., Exploring the Unrealized Potential of Computer-Aided Drafting, *Proceedings of CHI '96* (1996) 332–339.

[17] Carroll, J.M. and Aaronson, A.P., Learning by Doing with Simulated Help, *Communications of the ACM*, *31* (9) (1988) 1064–1079.

[18] Fisher, G., Lemke, A.C., Mastaglio, T. and Morch, A.I., The Role of Critiquing in Cooperative Problem Solving, *ACM Transactions on Information Systems*, *9* (3) (1991) 123–151.