THE WRITERS AND WRITING OF COMPUTER USER DOCUMENTATION:

A SOCIAL PERSPECTIVE

THESIS

Presented to the Graduate Council of the
University of North Texas in Partial
Fulfillment of the Requirements

For the Degree of

MASTER OF ARTS

By

Sheree C. Webb, B.A.

Denton, Texas
August, 1991

This thesis studies the writing of computer user documentation from a social perspective by examining the process of creating computer documentation and the role of documentation writers in the work place. This study consisted of in-depth interviews and observations of four writers of computer user documentation.

The thesis makes the following conclusions:

- Creating computer documentation is a social process that both influences and is influenced by the company in which it occurs.
- The documentation writers in this study are well-attuned to the needs of the users.
- Programmers and engineers are reluctant to provide information to the documentation writers.
- The documentation writers have a second-class role in their companies, a role that limits their ability to write effective documentation.
At first glance, writing computer user documentation would seem to be a fairly isolated process, involving only a writer and his or her computer or pen and paper. Closer inspection reveals that it is a social process. It involves social interaction between documentation writers and a variety of co-workers, including programmers, engineers, managers, and other writers. The process both influences and is influenced by the company or organization in which it occurs. To study this social process, researchers must, as Lester Faigley advises, move "beyond the traditional rhetorical concern for audience . . . to consider issues such as social roles, group purposes, communal organization, ideology, and finally theories of culture" (235-36). The purpose of this study was to look at documentation writing from a social perspective by examining the process of creating computer documentation and the role of documentation writers in the work place.

Previous Studies

In recent years, researchers have begun to study the relationship between writing and social and organizational contexts. They have found that social and organizational
contexts influence writing and that writing influences the context (Doheny-Farina, "Writing in an Emerging Organization" 160).

Social and Organizational Contexts Influence Writing

The conclusion that social and organizational contexts influence writing is supported by studies by Richard C. Freed and Glenn J. Broadhead and by Lee Odell. Freed and Broadhead studied the proposals of two similar organizations, a large management consulting firm and a large international accounting firm. They found that the cultures of the two firms influenced the proposals and the writers (162). For example, the culture of the accounting firm was permeated by caution, which was reflected in its proposals. This caution was especially evident in the benefits section of the proposals. At times this caution went so far that the benefits section was omitted altogether from the proposal (159). On the other hand, the culture of the management consulting firm considered benefits as a necessary concept. Therefore, its proposals not only included a persuasive benefits section but also discussed the benefits throughout the document (160). Thus, Freed and Broadhead concluded that the culture of the firm greatly influenced the writing of the proposals.

Odell, too, has found that the organizational context influences writing and writers. In his ethnographic study of workers in a state bureaucracy, he conducted discourse-
based interviews with a small group of supervisors and administrative analysts whose "principal tasks were to assess proposed legislation and to design procedures to implement legislation and agency policy" (250). By interviewing these workers about their memorandums, letters, and analyses of legislation, Odell concluded that the writers made decisions about their writing and their inquiry process based on the expectations and norms of the office in which they worked (251).

The Reciprocal Relationship Between Writing and the Organizational Context

Not only do the social and organizational contexts influence writing, but writing influences the organization. Studies by Stephen Doheny-Farina and by James Paradis, David Dobrin, and Richard Miller support this conclusion. Doheny-Farina conducted research for eight months at a newly-formed computer software company, where he studied the collaboration of a group of company executives as they wrote the 1983 business plan. His study showed that the writing of the business plan "both shaped and reflected the company's organizational context" ("Writing in an Emerging Organization" 165). For example, when the collaboration began, the president was the leader of the company and the principal writer of the first draft of the business plan. But when the company's structure changed to committee control, the committee members shared responsibility for
redrafting the business plan. Thus, the writing reflected the company's organizational context. On the other hand, as the company executives collaborated in the writing of the business plan, they began to challenge the president's authority. As the collaboration continued, the company implemented committee control. As Doheny-Farina points out, "the writing process helped to resolve the power struggle . . .," thus shaping the organizational context ("Writing in an Emerging Organization" 178).

Paradis, Dobrin, and Miller observed the writing activities of thirty-three engineers and scientists at the research and development division of Exxon. They found that the organization influenced writing in the document cycling process. In this process, the supervisors reviewed the documents of their subordinates and called for revisions. These required revisions reshaped the documents so that they fulfilled the company norms (294). During the document cycling process, Paradis, Dobrin, and Miller discovered, a conflict often arose between novice employees and their supervisors. In their documents, the novices wanted to emphasize their problem-solving process to show the soundness of their approach to a problem. The supervisors, however, wanted to see the results and how those results met the goals of the company (301). By calling for revisions in the novices' documents, the supervisors promoted the organizational norms. This study by Paradis, Dobrin, and
Miller also showed how writing played several roles in shaping the organization: it

- encouraged problem-solving (288)
- conveyed administrative information (292)
- stimulated ideas (298)
- helped to educate employees (298)
- played an important role in the employee review process (288)

Like the study by Doheny-Farina, this study showed the reciprocal relationship between writing and the organizational context.

The Social Perspective of Computer Documentation

Researchers have also begun to apply this social perspective to computer documentation and documentation writers. Studies by Barbara Mirel, by Richard Chisholm, and by Steve Anderson and Michael Kleine have examined the relationship between documentation writing and organizational context.

Mirel studied how in-house documentation and in-house documentation writers influence the organization in her case study of the development and implementation of an in-house accounting system manual. She found that the in-house documentation had another purpose besides its obvious purpose of instructing: it also helped "stabilize organizational life," which had become unstable because of the change to computerization (280). The documentation
helped the organization "adapt smoothly to computerization" (288). She also found that the documentation influenced the organization because it helped to establish a "consensus between users and programmers" about their roles in the changing work place (283). Documentation writers, too, influenced the organization because they acted as liaisons between users and programmers and as gatekeepers of information (283).

The studies by Chisholm and by Anderson and Kleine show how the organizational context influences writing, and they suggest ways for writers to influence the organization. Chisholm conducted a survey of documentation writers in which he asked them to describe the document cycle in their firm, to identify problems they had experienced, and to suggest solutions to those problems. The responses to his survey indicate three ways that the organizational context influences the writers and the documentation. First, the company's document cycle affects the writers and the documentation. If the company brings in the writers at the end or in the middle of the project, the contribution the writers can make to the project is seriously limited (302). However, if the company brings in the writers at the beginning of the project, the writers can contribute fully to the project, from design to sales (303). Second, managers often fail to plan and to schedule writing projects, leaving writers with insufficient time to write
effective documentation (305). Third, managers fail to promote collaboration between writers and programmers (306). When this failure occurs, writers cannot get the information they need to write effective documentation.

In response to the third section of Chisholm's survey, the documentation writers suggested solutions to the problems they frequently encounter. Their solutions illustrate how writers and documentation can influence the organization. For example, they suggest involving the writers at the beginning of the product development cycle (304), getting the managers to schedule adequate time for writing documentation (313), and encouraging an atmosphere of cooperation between programmers and writers (312). Implementing these suggestions would change the organization: the role of documentation writers would be upgraded so that they would be equal to programmers, writing would be viewed as part of the product development cycle instead of an afterthought to the product, and the documentation would be viewed as an integral part of the product instead of mere support for the product.

Anderson and Kleine observed similar problems when they were invited to be potential consultants at a large software corporation. The company's customers were not reading the documentation but instead were calling the company hotline for answers to routine questions. The company asked Anderson and Kleine to recommend how to motivate its
documentation writers to write usable user manuals. Anderson and Kleine found that the writers were fully capable of writing usable documentation, but that the company's assumptions about writers and writing prevented the writers from producing good documentation. For example, the company considered documentation to be of secondary importance and did not expect the customers to read the documentation (53). The company also viewed the documentation writers as second-class employees and viewed their work as merely clerical (55). Since they were viewed as second class, they were left out of planning sessions and were not allowed to deviate from the company's existing models of documentation nor to substantially change the documents given to them by the programmers (53-57). In this company, then, the organizational context had a negative influence on writers and writing.

Anderson and Kleine suggest solutions to this negative influence, such as educating the whole corporation about the valuable role of writers and documentation and creating an environment of collaboration among all components of the corporation (59-60). Such an environment would promote cooperation and equality between writers and programmers (61). Like Chisholm's solutions, these solutions, if implemented, would influence the organization.

These studies of the social perspective of documentation writing begin to explore the social and
organizational contexts of documentation writing. However, Mirel confines her study to one in-house manual, and Anderson and Kleine visited only one software company. Chisholm conducted a survey, giving him a general view of documentation writing but not an in-depth view of specific organizational contexts. Although these studies provide a good starting place for research into the social perspective of documentation writing, further research is needed.

Research Questions

To learn how documentation and documentation writers both influence and are influenced by their organizational context, I investigated these four research questions:

- What process do documentation writers go through to create computer documentation?
- Are documentation writers well-attuned to the needs of the users?
- How do documentation writers collaborate with programmers and computer engineers?
- What role do computer documentation writers occupy in their work place?

Methodology

To investigate these questions, I interviewed and observed four writers of computer user documentation. I chose three participants through contacts at the University of North Texas. I contacted the other participant through
the Lone Star Chapter of the Society for Technical Communication.

To strengthen the validity of the study, I used methodological triangulation, which Doheny-Farina and Odell define as a procedure for gathering data in which researchers use a variety of research methods to collect data from different sources (509). Based on suggested research methods in Jeanne W. Halpern's article "Getting in Deep: Using Qualitative Research in Business and Technical Communication" (33-34), I collected data using the following four research methods:

Phase 1: Conduct Initial Interviews

I interviewed the writers to collect background information about their education, their work experience, and their position at the company. I also asked them questions about the process of writing documentation such as the following: "Do you compose on the computer?" "Do you share the writing responsibility with others? If so, how?" "Do you have access to the end users of the documentation?"

Phase 2: Observe the Writers at Work

I spent half a day observing each writer performing normal daily activities at work. As the writers performed their usual activities, they explained briefly to me what they were doing. I recorded the time they spent on each activity and made notes about the activity. If possible, I attended any formal or informal meetings the writers had
with others. I also observed and took notes on the writers' work environments, noting the answers to questions such as the following: Do they have a private office or do they share an office? Is the work area quiet or noisy, relaxed or hectic? How do other employees respond to the writers?

**Phase 3: Study the Documentation**

I collected one or two manuals from each writer to examine their format, style, and organization. Using the manuals, I developed open-ended, discourse-based interview questions, adapted from the method described by Odell, Dixie Goswami, and Anne Herrington (223). In their discourse-based interviews, they studied documents written by an employee and then developed questions based on the documents.

**Phase 4: Conduct Discourse-Based Interviews**

Using the discourse-based questions, I interviewed the writers again. As we looked at their manuals together, I asked the writers why they chose a particular format, word or phrase, or component of the manual. In addition, I asked the writers the following questions: "How did you make the manual accessible and readable?" "What did you want to do, but didn't or couldn't? Why not?" "What did you try that didn't work?"

**The Writers' Backgrounds**

This section presents information about the writers' education, work experience, and work environments.
Susan

Susan has worked for almost two years full-time as a documentation specialist in the Management Information Systems department of the national headquarters of the American Heart Association in Dallas, Texas. Including her previous work experience, she has had three years of experience writing documentation. She was originally hired by her previous employer to write manuals not because she knew how to write well but because she knew the computer system. Therefore, according to Susan, her first manuals were poorly written. She has a Bachelor's degree in business administration and is currently taking night classes to obtain a Master's degree in technical writing at the University of North Texas.

Susan writes computer user documentation for software programs developed at the American Heart Association. Her manuals are used by non-technical employees at both the national headquarters in Dallas and the local affiliates in other cities and states. The users' familiarity with computers varies--some have extensive computer experience while others have never used a computer. Therefore, she has to write so that even the least experienced users can follow the manual. Susan also writes on-line documentation and edits the departmental newsletter.

Susan's work area is a cubicle in a large room with the other members of the Systems Development division of the
Management Information Systems department. This division consists of programmers; one other documentation specialist, who is Susan's supervisor; and the document coordinator. In her cubicle, she has a computer terminal, an L-shaped desk, and shelves for storage. Although her work area is fairly quiet, noise carries easily; and she can hear other employees talking, shuffling papers, and typing. The cubicle gives her some privacy, but people passing by can be distracting. Susan says that she has learned to block out distractions and just keep writing. Susan's work area is similar to the work areas of the programmers and other documentation employees.

Dave

Dave worked part-time for Bell Northern Research, the research and development division of Northern Telecom, in Dallas, Texas, where he was the only technical writer. He worked there for one year. This position was Dave's first job as a technical writer. Since our interviews, Dave has moved to another company. He has a Bachelor's degree in English composition and is currently working on his Master's degree in technical writing at the University of North Texas.

Dave worked on one manual while he was employed at Bell Northern Research: a software manual for engineers and programmers. He also wrote a newsletter column and some public relations material.
Dave's work area was a partitioned section of a long table divided into five sections, similar to a row of library carrels. Since Dave and the other employees who worked on this row were part-time, cooperative education students, this row was known as "Co-op Row." In his work area, Dave had a computer terminal connected to the main frame, a personal computer, and an overhead storage compartment. When I visited Dave, "Co-op Row" was deserted except for Dave. Since his work area was in the corner of the room, he could not look up and see any other employees. His work area was very quiet; in fact, it was almost isolated.

Sarah

Sarah has worked full-time for one-and-a-half years as a technical writer for Dallas Systems Corporation, a small software design company, where she is the first and only technical writer. Before she began working for the company, the programmers wrote the documentation. She has written over thirty-three manuals during the eight years she has been writing computer documentation. She has a Bachelor's degree in computer science and had planned to work as a programmer. However, since she could not find a job as a programmer, she began working as a technical writer. She found writing computer documentation more rewarding than programming and decided to make writing her career. She has
won awards the last two years from the Society for Technical Communication for her hardware documentation.

Sarah writes computer documentation for integrated hardware and software programs developed at Dallas Systems Corporation. These programs are used by its clients in their warehouses to assign and monitor the work of warehouse employees. In addition to writing computer documentation, she is also responsible for documentation management, including ordering tabs and binders, deciding how many copies of a manual need to be printed, and keeping a record of which client has which version of each manual. Sarah also evaluates equipment to decide which software packages and which copier the company should buy. She may soon begin working on a computer-based training program.

Unlike Susan and Dave, Sarah has a private office. Everyone at Dallas Systems Corporation has a private office. In her office she has both an IBM-compatible personal computer and a Macintosh personal computer. She also has a desk, shelves, a filing cabinet, and a bulletin board. When I visited her, her floor, desk, and filing cabinet were covered with various versions of an update she was sending out to customers. Although Sarah has a private office, it was not quiet when I observed her. Her telephone intercom frequently announced telephone calls to other employees, interrupting Sarah's concentration.
Bob

Bob has worked full-time as a technical writer for Texas Instruments in Plano for seven years. Including his previous experience, he has worked as a technical writer for eleven years. Bob "fell into" technical writing; he did not plan to work as a technical writer when he finished college. He always enjoyed writing, and he enjoyed learning how things worked. In fact, he did best at writing about how things worked. At one of his jobs, he worked with computers and liked learning about them. His next job was with a computer product manufacturer where he learned more about computers. Then he finally began writing computer documentation, enjoyed it, and has been writing it ever since.

He has a Bachelor's degree in speech and journalism with several electives in electronic engineering. He is currently taking night classes toward his Master's degree in communication with a minor in technical writing. His education also includes several short courses offered by Texas Instruments to its employees. Although a few of the courses apply directly to technical writing, most of them are designed for programmers and engineers. Bob takes these courses too because the more he can learn about computers and the systems, the better he can communicate with the programmers and engineers from whom he has to gather information. Bob has also attended several seminars offered
by the Society for Technical Communication to learn about the latest technology and theory in technical writing.

Bob writes computer documentation for software programs developed at Texas Instruments. The company uses its own software to control human resources records such as payroll, benefits, and vacation and to control fiscal records such as accounts payable and receivable. The documentation is used mostly by non-technical employees at Texas Instruments locations worldwide. Bob also edits a monthly newsletter for members of his group and writes computer-based training, an on-line training program that teaches the users how to use Texas Instruments' computer systems.

Bob is a member of the group responsible for developing the software for the financial departments such as accounts payable and receivable, general ledger, and tax and treasury. His group consists of seventy to eighty employees, mostly software engineers and programmers. The group also includes two other technical writers with whom Bob shares the responsibility for documentation. Outside of Bob's group, Texas Instruments employs approximately twenty-five other technical writers at its Plano location.

Like Susan, Bob has a cubicle in a large room. The walls of the cubicle are carpeted so that sound does not carry. Overall, the environment is quiet with few distractions. Bob has two personal computers in his cubicle, one of which is connected to the mainframe. He
also has an L-shaped desk, shelves, and storage compartments. The other two technical writers have cubicles on either side of Bob's so that he can easily communicate with them.

The Social Perspective

This section explores the social perspective of documentation writing by discussing the process these documentation writers go through to create their manuals, their awareness of the users' needs, and their interaction with the programmers and engineers. It also discusses the role of documentation and documentation writers.

The Process of Creating the Documentation

Writing computer documentation is a social process, involving interaction with programmers, engineers, and managers. Although the specific procedure varies somewhat from writer to writer, the documentation writers in my research overall follow similar steps to create computer documentation:

1. Gather information by interviewing programmers, engineers, and managers and by running the program.
2. Decide on the design, audience, and purpose of the documentation.
3. Write a draft of the documentation.
4. Have the documentation checked for accuracy and fill in missing information.
5. Revise.

6. Review the final copy with the project leader.

The following sections first describe the work situation for each writer and then present in detail the procedure each writer follows to create the documentation.

**Susan.** Susan and her supervisor do not write collaboratively. When Susan is assigned a manual to write, she writes it on her own: her supervisor does not review her work or share the writing of that manual.

Susan conducts user-testing of the manuals and makes changes based on the tests. For example, she designed one manual with "How-To's" in the back, but when she conducted user-testing, she found the design would not work because the users had to flip back and forth from the front to the back of the manual. She also sends drafts of the manuals to the users so that they can give her review comments. However, she seldom gets any review comments and suspects the recipients do not use the manuals; instead they ask someone else in their office how to use the software. Susan comments, half-jokingly, that if she could come up with an index card instead of a 1,000-page manual, they would use it more. To figure out how to get the users to use the manuals, Susan is currently studying the minimalist ideas of John Carroll, who believes if writers give the users less, they will use the manual more (150).
The following example illustrates the typical process Susan goes through to write a manual. Susan was assigned to write a manual for a subscriptions software program developed at the American Heart Association. This program keeps track of and fills the subscriptions to the journals published by the American Heart Association. When Susan became involved in the project, the main programmer had already interviewed the users about what they needed from the program and had begun developing the program and building prototype screens. Susan interviewed the main programmer and the business specialist to gather information about the program and the manual. She also gathered information by running the software program. From the notes she took while gathering information, she wrote a draft of the manual on the computer, using the Mass 11 word processing program.

Initially, Susan had a six-month deadline and a 350-page limit. Because of the complexity of the program, the deadline was repeatedly postponed. As the software continued to grow, so did the length of the manual. Susan continued to add the new material to the manual.

Susan went through several designs in ten months, trying to combine a reference manual and a users' manual. She finally had a design she thought would work, but she found out through user-testing that the design did not meet the users' needs for easy access to the information.
Although she asked the users to contribute their ideas about designing the manual, they gave her none. She, therefore, came up with another design on her own. She thinks the current design works well and has not received any negative comments from the users. However, as mentioned previously, she suspects they have not used the manual. In fact, during a training session, the users repeatedly asked questions that they could have looked up in the manual.

Next, Susan gave a fairly complete draft to the business specialist for a technical edit; he also checked it for accuracy. Based on his remarks, she revised the manual and then distributed the latest draft of the manual to the users for review comments, but again did not receive any comments. Then Susan's assistant, the document coordinator, edited the manual for consistency and correctness. Next, Susan compared the program screens to the manual screens to check for changes that the programmers might have made without telling her. The final manual was supposed to be finished in mid-November, but she was still working on it in late November, and she believed that the deadline would get pushed back again. At that time, Susan had been working on the manual for nearly two years, and it had grown to 1,000 pages.

**Dave.** Like Susan, Dave wrote on his own, not as a part of a writing team. He was a member of the design team assigned to developing TRES, the software program for which
Dave wrote the manual. This program simulates telephone calls, thereby automating the testing of the company's software for its telecommunication switches.

Dave worked on the TRES manual for the entire year he was employed at Bell Northern Research. To write the manual, he went through the following process. When he started working at Bell Northern Research, he had little computer experience beyond word processing; yet he was assigned to write a technical software manual for an audience of engineers and programmers. To begin work on the manual, he read and reread the programmers' documentation about the program. He did not understand much of what he read and had to ask the programmers to explain the material to him. He often could not understand the programmers because they had trouble explaining the program to him in language he could understand.

Dave also had access to the software program itself; but since he had received no formal training on the software, he had trouble understanding how to work the program. Finally, he did get some informal training on the basics of the software. From what he could learn, Dave divided the manual into major sections, then subsections, continuing to fill in information until he had enough information to begin writing the text. Dave wrote the manual on the computer using Microsoft Word. After he had begun designing the manual, he found that the company had a
standard template for manuals. Dave then used the template to design and organize the manual.

As he continued to work on the manual, Dave encountered further problems. First, he had trouble tracking down the answers to his questions. Since often only one person knew a particular answer, he had to ask several people until he found the right person. Second, the software changed constantly, causing parts of the manual to become obsolete as Dave wrote them. At times, Dave finished writing a section of the manual only to discover that section was no longer needed. Dave continued to work on the manual, editing the sections he had already written and filling in missing information. Third, when Dave asked the technical members of the team to review the manual for accuracy, he had trouble getting review comments. Instead, they often just said, "Looks good." Dave suspects they did not read it. Once Dave had gotten as many review comments as he could, he revised the manual once more and then sent it to his supervisor for approval.

Sarah. Sarah usually works under the pressure of time constraints. She writes the documentation for the programs designed by all thirty of the company's programmers. She works on several projects at one time. When I visited her, she was updating three manuals, writing two others, and preparing to begin work on three others. She often stays at
The following paragraphs describe the usual procedure she follows to create computer documentation. When the programmers have finished a program, except for revisions, Sarah is assigned to write the documentation. Sometimes the company has already sold the program to a customer and needs the documentation within a month. Sarah first has to learn how the software runs or how the hardware is assembled. For software, she runs the program herself to see how it works and also has the designer explain the program to her. To see how the hardware is assembled, she watches somebody put it together. Next, she talks to the project manager to find out who will be using the software or hardware and what tasks they will be using it for. She takes notes as she is learning about the hardware and software and the users. Using her notes, she then writes the documentation with pen and paper, sometimes writing a section at a time, sometimes writing the entire manual at once. Then she enters the information on her computer, editing as she types. After editing, she makes sure she has organized the manual logically and then adds the cross references.

Next, she tests the manual against the software to see if it works the way the manual says it does. As she tests it, she tries to put herself in the users' place and think from their point of view. During this testing, Sarah finds...
out what the programmers have changed. According to Sarah, they always change the software without telling her.

Once she has updated the documentation to include the changes, she formats it using the **Ventura** desktop publishing package. Since the company has no standard format, Sarah makes the formatting decisions. Although she uses a fairly consistent format from manual to manual, she may change some aspect of the format if she finds a better way to do it. Therefore, the manual format has evolved during the time Sarah has worked for the company.

After formatting, Sarah gives a copy of the manual and a list of questions about its content to the project manager or the programmer, who checks the manual for accuracy and answers the list of questions. Sarah often has trouble getting the programmers to read the manual or to answer her questions. According to Sarah, she has to "hold their hands" through the review process; otherwise, they will return the manual to her with the comment, "It's fine."

After the manual is complete, Sarah has her assistant proofread it for typographical, spelling, and usage errors. She sends the manual to the printer, orders tabs and binders, and supervises the final production of the manual.

**Bob.** Although Bob shares the responsibility for writing documentation with Glenda and Jean, the other two technical writers in his group, he does not write collaboratively with them. They usually work individually...
on separate projects. Occasionally, if the project is very large, two technical writers will work on it, but they divide the project up and each works on his or her section individually. For example, Bob and Glenda are both currently working on a computer-based training program since it is a large project. However, they have divided the sections between them so that they can work independently. According to Bob, the technical writers in his group are more likely to collaborate in the early stages of information gathering rather than in the drafting stage.

Bob's biggest problem in writing documentation is time. The technical writers do not usually have enough time to write the documentation. When Bob's group members plan a project, they do not allow enough time for documentation. As the project continues, the writers are usually left with even less time than planned to write the documentation.

Bob usually completes the following procedure to write computer documentation. Bob's supervisor assigns him to the project after the software engineers have gone through multiple design and specification meetings and the programmers have begun coding the program. Bob contacts the project leader and sets up a meeting to discuss the general information about the project such as the users, the due date, the name of the software, and the name of Bob's contact person. Then Bob estimates how long the documentation will take to produce and makes an outline of
tasks for himself. He gathers information about the software by interviewing the programmers and his contact person. He also usually has access to the software program itself so he can use it to find out how it works.

From the notes he has taken during the information-gathering process, Bob starts writing on the computer. Since he usually needs further information, he imbeds questions to himself and to the contact person into the documentation. He conducts more interviews to fill in the holes and then adds that information to the documentation.

When he has a fairly complete working draft of the document, he sends it to the project leader for an accuracy check. Sometimes the project leader sends it to the programmers so they too can check it for accuracy. The project leader then returns the draft to Bob, usually covered with comments written in red ink. Occasionally, the project leader returns it, saying, "It's fine," and Bob knows she or he has not read it.

Based on the review comments, Bob revises the document and, if time permits, sends it to an expert user or user trainer, a person very familiar with the software, to find out if the system works the way the documentation says it does. Bob revises the documentation again based on the user's comments and then reviews this final copy with the project leader. Together, they make the final decisions.
about the editing, style, and fonts. Bob then formats the manual using the Ventura desktop publishing program.

Writers' Awareness of the Users

My research indicates that these four writers are well attuned to the users' needs although their organizational contexts do not always allow them to use what they know about the users' needs to create effective documentation. This section discusses how the writers make their documentation accessible and readable, how the writers express their concern for meeting the users' needs, and how the organizational context hinders the writers' efforts to meet users' needs.

Accessibility of the Documentation. Based on the documentation itself and the discourse-based interviews, I categorized the ways the writers make the documentation accessible and readable into

1. format
2. writing style
3. content and organization

Format. The format of the documentation of all four writers is remarkably similar. Their documentation looks neat and professional because it has been created using word processing or desktop publishing programs. Their documentation uses left-hanging headings. In addition, all four writers use the following formatting techniques that experts agree make documentation accessible:
ample white space (Benson 38; Debs 18; Ramey 155; Redish et al. 146)
headers with the chapter or section name at the top of the page (Redish et al. 146)
headings and subheadings (Debs 18)
ragged-right margins (Benson 38; Debs 19)
displayed lists for step-by-step instructions (Debs 18)

In addition, some of the writers make the documentation even more accessible by
using tabs and dividers between sections (Ramey 154; Redish et al. 146)
putting the chapter name and section name on the front and back of the tabs (Ramey 154)
positioning the page numbers and headers on the outer half of the page so that the user can easily find the information by flipping through the pages (Redish et al. 146)

Writing Style. All four writers strive for a clear, concise writing style—a style that conveys the necessary information in as few words as possible. To achieve this goal, the writers use words familiar to the users (Selzer 79), simple sentences (Selzer 79-80), and action verbs in the imperative mood (Lannon 332).

On the other hand, the formality of the writing style varies from writer to writer and from document to document.
For example, Bob uses a friendly, conversational style for his computer-based training program. In the introduction to one section he tells the users, "During the next few minutes we'll be talking about the specifics of the program--its purpose and how it works." For this project, Bob wants to convey the sense that he is talking in person to the users. At other times the writers must use a more formal tone such as this example from one of Sarah's manuals, "With this system you can access information on work to be performed, the status of work assignments, and the productivity of each employee."

The wording of the headings and subheadings also varies from writer to writer. Two of the writers used conventional, system-oriented headings such as "Assignment Adjustment," "Data Transfer," and "KPA/PC Reporting." The other two writers used task-oriented headings such as "How to Enter Your Order," "How to Use the Function Keys," "How to Add an Associate," and "Coding a Test Case." Experts agree that task-oriented headings are more helpful to users (Benson 38; Flower et al. 54-55; Ramey 145; Redish et al. 144).

Content and Organization. The information the writers chose to include demonstrates their awareness of the users. All four writers included the following components, which experts have found helpful to users:

- tables of contents (Ramey 154; Redish et al. 146)
- overviews of each chapter (Redish et al. 143)
- prefaces or introductions to the documentation (Redish et al. 143)
- sample screens (Debs 18)

In addition, three writers included indexes (Ramey 153; Redish et al. 146) and error messages. The error messages, however, would have been more helpful if they had told the users what to do instead of simply telling what the error message meant (Carroll 150; Mirel et al. 81). Two of the writers also included quick reference cards (Redish et al. 145) and situational examples (Charney et al. 63; Flower et al. 55; Mirel et al. 80).

How the writers organized the documentation also reflects their awareness of the users. All four writers attempted to make their documentation modular, that is to write it so that users could refer to one section of the documentation to perform their tasks without having to refer to another section or to the rest of the documentation. Susan, in fact, is planning her next manual so that the users can physically separate it into sections by task and use each section separately.

The writers also try to organize the documentation consistently so that the users can find similar information in the same place from chapter to chapter and from section to section (Benson 36; Redish et al. 146). For example, each section of Susan's subscriptions manual begins with the
screen and has instructions on how to access the screen directly below each screen.

Three of the writers organized their documentation by task, an organization that is more user-friendly (Flower et al. 54; Ramey 150). The one manual organized by screen was Dave's TRES manual, intended for technical users. According to Dave, this organization was appropriate for the users because they would use the documentation to look up a screen and its field definitions, not to look up how to perform a particular task.

**The Writers' Concerns for Meeting the Users' Needs.** My observations of and conversations with these writers also indicate their concern for meeting the users' needs. All four writers attempt to look at the documentation from the users' point of view. Of the four writers, Susan works most extensively with the users, perhaps because she has easy access to her in-house users. She conducts user-testing of the documentation several times, gives the users numerous opportunities, both oral and written, to voice their opinions of the documentation, and includes a document evaluation sheet in the final version of the documentation. Despite her efforts, however, Susan often receives few comments or suggestions from the users. To find out more about the users' needs, Susan surveyed them to find out what parts of the documentation they considered most and least helpful. They rated tables of contents and indexes as most
helpful and overviews and introductions least helpful. Susan plans to take these survey results into consideration as she writes future manuals. Susan would also like to make other changes in her future manuals so that they better meet the needs of the users. She wants to include more error messages and information about how to correct errors. She would also like to include more situational examples and more information explaining why the users are performing a certain procedure.

Although Bob does not have time to conduct user-testing of his documentation, he does attempt to meet the users' needs in other ways. First, he asks an expert user to review the documentation to see if it accurately describes the program. Second, he includes maps of the documentation to guide users to the place they need to look (Benson 36; Redish et al. 145). For example, he may guide users by informing expert users to go to a certain page, intermediate users to go to another page, and novice users to go to yet another page.

In Dave's case, the programmers and engineers were the users. He gave them review copies of the documentation so they could give him suggestions. However, like the other documentation writers, he suspects they often did not read the material.

Sarah has the least contact with the users since she has no in-house users. She does, however, attempt to meet
the users' needs. She tests the documentation against the program by looking at the documentation from the users' point of view and by running the program to see if it does what the documentation says it should. She also organizes the documentation in the order that the users will need the information.

**Constraints of the Organizational Context.** Overall, the writers are concerned about the users' needs and attempt to meet those needs. However, their organizational contexts do not always allow them to meet the users' needs. Susan, in particular, finds it difficult to put what she knows into practice. For example, when she was writing the subscriptions manual, the Management Information Systems department allowed her to write only one manual. Susan would have preferred to write a reference manual and a users' manual, but instead she had to combine the information into one 1,000-page manual. As mentioned previously, Susan does not think that the users use this manual: the length itself is intimidating to users, and since it is bulky and unwieldy, it is difficult to use.

Susan had two other ideas for the manual that her department did not allow her to implement. First, she wanted to make the manual task-oriented so that it would be more helpful to the users; but her department required it to be system-oriented. Second, she wanted to make the manual a training guide since she believes that is what the users
need. According to Susan, they use the manual as they are
learning the system and then never look at it again.
Therefore, they use it as a training guide, not as a
reference manual. However, the department required that she
write a reference manual because that is the way it has
always been done.

**Interaction With Programmers and Engineers**

These documentation writers need to work closely with
the programmers and engineers since these technical
employees know how the software works and are often still
changing the software even while the documentation is being
written. However, my interviews and observations indicate
that the programmers and engineers do not cooperate with the
documentation writers. This problem is two-fold. First,
the documentation writers have trouble getting the
programmers and engineers to answer their questions and
check the documentation for accuracy. All four writers in
my research commented several times that most programmers
and engineers were reluctant to answer their questions. For
example, when the writers send a review copy of the
documentation to the programmers or engineers for an
accuracy check, they often return the documentation with the
comment "Looks good" or "It's fine." The documentation
writers agree that when they get this response, they know
the programmers and engineers have not read the
documentation. Another common example occurs when the
documentation writers ask the programmers and engineers questions about the program: they often tell the writers to ask someone else. As Bob puts it, to get their questions answered, the documentation writers have to "ask the right person the right question at the right time in the right way."

Second, the programmers often change the software program without informing the documentation writers of the changes. All four documentation writers agree that the programmers frequently change the software without telling them. In fact, this omission happens so often that the documentation writers routinely check the software for any changes. When I observed Bob, he was going through the program screen by screen to compare the program screens with the sample screens in the documentation he had written. He has learned through experience that the programmers will have changed some of the screens without telling him. Likewise, when Sarah has written a draft of the documentation, she checks it against the program to see what has been changed. The documentation writers also hear "through the grapevine" that screens have changed, and they have to check the software program for these rumored changes.

According to Bob, the programmers do not maliciously withhold information from the documentation writers; instead, they simply do not realize the importance of
documentation. This attitude is a holdover from the time ten to fifteen years ago when documentation was simply an afterthought—not particularly necessary because only experts used computers, and they already knew how the software worked (Anderson and Kleine 54). Although the users have changed and now include many more lay people than computer experts (Brockmann 18), the programmers' attitudes toward documentation have not changed. According to the writers in my study, most programmers still view documentation as a bothersome afterthought to creating the software.

The Role of Documentation and Documentation Writers

The programmers' and engineers' attitude toward documentation is only part of the overall picture. My interviews and observations indicate that the four companies in my study regard documentation and documentation writers as second class, a conclusion which supports the findings of Anderson and Kleine (55). For example, management does not allow enough time for documentation in the planning stages of a project. Then as the project progresses, the documentation stage often has even less time than originally planned. Management's attitude toward scheduling time for documentation is typified by an experience Bob recently had. Bob attended a meeting where the manager in charge of a proposed software program called Project Planner presented the proposed software with carefully designed transparencies.
and described in detail how the program would work: the user would simply fill in the blanks with the requested information, and the program would figure how much money, time, and personnel would be needed for the project. The program, however, failed to include a blank for documentation. Further, the manager in her presentation neglected to mention documentation; so the proposed software, which was supposed to accurately predict the cost, time, and personnel required for a project, did not take into account the cost, time, and personnel required for the documentation of that project.

The companies in my study also give documentation writers low priority. During my research, I heard about and observed several examples of documentation writers' being treated as second-class employees. The most telling of these examples occurred when I twice observed documentation writers who could not access the computer file they needed. Susan could not access the documentation file—a file she had created—to make some changes to the documentation. She had to wait for a programmer to return her call and inform her that the password to the file had been changed, without anyone's notifying her—the person responsible for the file—of the change. Similarly, Sarah could not log-on to the software program she was documenting to check for changes in the program. After two futile telephone calls to find out why she could not log-on, she walked down the hall
to the office of one of the programmers. When she explained to him that she could not log-on to the program, he tersely replied, "So?" After she explained further, he finally told her that the password had been changed, without her being notified. As we walked back to her office, she commented, "They never volunteer anything. I have to pry it out of them like Sherlock Holmes."

Sarah had another experience that illustrates the low priority of documentation and documentation writers. Her company hired an assistant to help her with the documentation, an assistant she badly needed since Sarah comprises the entire documentation department. However, the assistant is required to do so much secretarial work for the programmers that she has little time left to help Sarah. Yet since she supposedly has an assistant, Sarah has been assigned even more work.

Dave's experience, too, illustrates the low priority of documentation and documentation writers. When Dave began working at Bell Northern Research, he had little experience with computers; yet he was assigned to write a manual for a technical audience about a complicated software program. He was given no formal training on how to use the software he was documenting. If the company wanted effective documentation, it would have either hired an experienced documentation writer or given Dave the training he needed.
Bob agrees that documentation writers are second-class employees. For instance, when a project at Texas Instruments has been especially successful, the programmers involved receive a financial reward. However, the documentation writers involved are overlooked and receive no reward. Their contribution to the project is not recognized or regarded as valuable. Bob says too that programmers and programming are the first priority at Texas Instruments. Although documentation writers usually create documentation the way they want, the programmers have the final authority to decide if it is right. Usually the programmers just check the documentation for accuracy, but occasionally they insist on changes in the appearance or style of the documentation. Sometimes, Bob admits, he has had to write or format documentation the way a programmer told him to even though Bob knew it was poorly written or formatted.

Susan too sometimes has to write or format a document ineffectively because of requirements of the Management Information Systems managers. They have their way of writing documentation, and they do not want to change. For example, they want system-oriented documentation because that is the way they perceive the program. But Susan says the users need task-oriented documentation. Also, the Management Information Systems department wants reference documentation because that is the way it has always been done. Susan, on the other hand, believes they need to write
training guides because that is how the users use the documentation. Through experience and technical writing classes, Susan has learned better ways to write the documentation, but the managers do not want to hear about it. The last time I talked to her, Susan was frustrated by the unwillingness of the Management Information Systems managers to listen to her suggestions. She commented that most companies have their set structure and technical writers are near the bottom of that structure.

According to the writers I interviewed, management does not understand why documentation writers want to be involved in the project as it is developed; they think the documentation writers need to become involved only near the end of the project—after the "real" work is finished. Management's attitude toward technical writers is summed up in a company joke that Dave told me:

Question: How many technical writers does it take to change a light bulb?

Answer: One, as long as there's a programmer to tell her or him how to do it.

Conclusions and Implications

From my research, I draw the following conclusions about the social perspective of documentation writing:

- Creating computer documentation is a social process that involves interaction between documentation writers and a variety of co-workers,
including programmers, engineers, managers, and other writers.

- The documentation writers in my study are well-attuned to the needs of the users although the constraints of the organizational context limit their ability to apply their knowledge.

- Documentation writers have trouble collaborating with programmers and engineers because the programmers and engineers are reluctant to answer questions and provide information.

- Documentation writers have a second-class role in their companies, a role that limits their ability to contribute to projects and to write effective documentation.

My research has implications for companies in which software documentation is written and for the education of documentation writers.

**Implications for Companies**

Companies need to recognize the value of documentation and documentation writers. For instance, they need to understand that without the documentation, most users cannot use the hardware or software (Chisholm 309). By enabling users to use the product, the documentation creates the need for the product. Similarly, companies need to recognize the central role documentation writers play. Documentation writers do much more than simply translate technical
material into lay terms. Instead they sift through the technical information that they gather from programmers, designers, and engineers to create documentation appropriate for the audience and purpose. Documentation writers provide the vital link between the product and the users.

If companies recognize the value of documentation and documentation writers, they will involve writers from the beginning of the project, when they can fully contribute to the project (Baker 320-21). Writers can offer valuable advice about designing the software, marketing the product, and training users (Chisholm 304). For example, Susan had several insights into what users wanted from a software program. She told me that users do not want good documentation; they want no documentation. In other words, they want an intuitive program that they can use without documentation, a program with self-explanatory commands and no blank screens. Yet since Susan's company does not assign her to a project until it is well underway, the project leader does not consider her insights.

To produce effective documentation, companies must encourage collaboration between technical employees and documentation writers (Anderson and Kleine 61; Chisholm 311-12). My research shows how the reluctance of the technical employees to provide information to the writers hindered the writers' efforts to produce effective documentation: if the documentation does not contain accurate information, how can
it be effective? Yet technical employees will remain reluctant to collaborate with "second-class" documentation writers until the companies recognize the value of writers and regard them as equal to the technical employees.

Implications for Education

My research shows the major role of communication and interpersonal skills in gathering information for computer documentation. Documentation writers need to learn how to develop and improve these skills. University courses in technical writing should include instruction in communication and interpersonal skills. As Odell points out,

We need to give students frequent opportunities to practice the interpersonal skills that will enable them to function effectively in a dialogue or in a group discussion. It will not be enough to ask students to work in groups; we will have to make sure that they know how to do so effectively.

(278)

In addition to universities, the Society for Technical Communication and companies with documentation writers should offer seminars or mini-courses to help documentation writers develop their communication and interpersonal skills.

"How to" books on writing computer documentation should also educate writers about gathering information from
programmers and engineers. Currently, such books ignore or minimize this vital step. For example, R. John Brockmann's *Writing Better Computer User Documentation* begins the process of documentation with the assumption that writers already have all the information they need. Even his chapter on reviewing the documentation ignores the possibility that writers will have trouble getting review comments from programmers and engineers. Brad McGehee, on the other hand, in *The Complete Guide to Writing Software User Manuals* briefly discusses collaborating with programmers but mistakenly advises readers that maintaining an open line of communication between the programmer and the writer is "usually not a problem if they both work at the same location . . ." (8). Herman Holtz in *The Complete Guide to Writing Readable User Manuals* includes one paragraph on interviewing programmers and engineers, admitting, "This is often a relatively difficult task. . . ." (52). To overcome this "relatively difficult task," he advises readers to use this step as a last resort (52), an unrealistic solution for most documentation writers since programmers and engineers are often their main source of information. Jonathan Price in *How to Write a Computer Manual: A Handbook of Software Documentation* briefly discusses gathering information from programmers and engineers (41). However, he does not acknowledge that they may be reluctant to answer questions and provide
information. These books fail to address one of the largest problems documentation writers have on the job (Chisholm 306).

**Further Research**

My study shows the reciprocal relationship between social and organizational context and documentation writing: documentation writing both influences and is influenced by the social and organizational context. Future research should continue to study the social perspective of documentation writing. In addition to interviewing documentation writers, researchers may want to interview programmers and engineers, managers, and users. Such interviews would give us a more rounded view of the role of documentation and documentation writers.
WORKS CITED


