Journal of Business and Technical Communication

http://jbt.sagepub.com

Creating Procedural Discourse and Knowledge for Software Users: Beyond Translation and Transmission

Marjorie Rush Hovde

Journal of Business and Technical Communication 2010; 24; 164 originally published online Nov 23, 2009;
DOI: 10.1177/1050651909353306

The online version of this article can be found at:
http://jbt.sagepub.com/cgi/content/abstract/24/2/164

Published by:
SAGE
http://www.sagepublications.com

Journal of Business and Technical Communication
24(2) 164-205

DOI: 10.1177/1050651909353306
http://jbtc.sagepub.com

Creating Procedural Discourse and Knowledge for Software Users: Beyond Translation and Transmission

Marjorie Rush Hovde[1]

Abstract

Although most theorists agree that discourse creates meaning, they have not adequately described how this process emerges within the creation of procedural knowledge. This article explores how technical communicators in diverse settings based discourse decisions on their knowledge of (a) users, (b) organizational image and constraints, (c) software structure and features, and (d) genre conventions in order to create communication artifacts designed to help users develop procedural knowledge. The transformations in which they engaged indicated that these technical communicators were skilled in forming images in these four areas and then using these images as they created meaning in procedural discourse. In this process, they moved beyond merely translating or transmitting technical knowledge.

Keywords

procedural discourse, observational study, technical communication, workplace communication, manual creation

[1] Indiana University Purdue University Indianapolis

Corresponding Author:
Marjorie Rush Hovde, Indiana University Purdue University Indianapolis, ET 324F, 799 W. Michigan St., IN 46202.
Email: mhovde@iupui.edu

Theorists have long argued that language constructs meaning (Farrell, 1976; Simon, 1990; Smart, 1999b; Spivey, 1997; Subbiah, 1997; Wells, 1986; Winsor, 1989), leading to an understanding of discourse as a "formative principle" (Backman, 1987, p. viii). This understanding has had important effects on the field of discourse studies, taking it beyond formal, textual concerns to cover a broad range of inquiries. But the claim that discourse creates meaning does not seem to have filtered into the consciousness of many creators of user documentation or that of their colleagues. Technical communicators find themselves involved in a "complex network of knowledge-making activity" (Smart, 1999a, p. 227), but colleagues in that network may still believe that technical communicators merely "transmit" or "translate" information (Slack, Miller, & Doak, 1993), neither of which seems consistent with theorists' views of the relationship between knowledge and discourse. Unfortunately, the perception prevails that technical writing concerns only styling and arranging text and that technical communicators have no role as authors who create meaning.

This limited view of their roles leads to unfortunate consequences for both technical communicators and users of technology. To act professionally, technical communicators need to understand and develop their expertise as meaning makers. A richer picture of how technical communicators create meaning within discourse through "both interpretation and reconstitution of the materials" (Simon, 1990, p. 22) or through "articulation" and "re-articulation" of meaning (Slack et al., 1993) will help to build a better professional image for technical communicators, who, upon seeing themselves as authors, may develop enhanced professional self-esteem.

An improved professional image is not an end in itself, however; the ultimate benefit of seeing technical communicators as meaning makers is that such an understanding can improve the usability of procedural discourse for end users. Effective discourse accompanying a new technology can empower users as they learn to employ that technology. In addition, the quality of such documentation affects users' judgments about a technical product (Smart, Madrigal, & Seawright, 1996). Furthermore, well-designed documentation can ameliorate "information liability" problems with technical products (Smith & Shirk, 1996). But if technical communicators are not seen as authors who create new meaning, they may be denied access to organizational resources that would allow them to create documentation that is most likely to be usable and useful for its intended audience.

Overall, well-designed user documentation serves to guide users' behavior in using technology to achieve their goals. If we believe that user documentation serves an "exegetical" function in that it can "interpret for the

lay public the meanings, applications, and procedures by which expert products ... are integrated into the behavioral flow of society itself" (Paradis, 1991, p. 256), we need to consider seriously the process of making meaning within processes of shaping user documentation. Technical communicators need to "apply rhetorical strategies to achiev[e] operational coherence and simplicity" (p. 259) in technical documentation that is intended to assist users in gaining procedural knowledge. If technical communicators and their colleagues are aware that technical communicators are making meaning, that awareness may enhance the circumstances in which technical communicators design procedural discourse for users.

Although scholars have engaged in theoretical discussions about how discourse creates knowledge, we lack observational studies that explore the factors that technical communicators consider as they create procedural knowledge materials for end users. To address this need, I have drawn on observational data from two studies of technical communicators to argue that these individuals created meaning in their discourse through integrating their knowledge of (a) users, (b) organizational image and constraints, (c) software, and (d) genre conventions. In the process of creating meaning, these technical communicators employed these four factors as they transformed declarative technical knowledge into discourse designed to assist users in employing technology. In the following sections, I explore differences between declarative and procedural knowledge, provide perspectives on technical communicators as creators of procedural knowledge, and examine contextual factors that may shape their discourse decisions. Subsequently, I describe my research methodology, analyze the findings, and conclude by discussing the implications of these findings for technical communication theory, practice, and research.

Differences Between Declarative and Procedural Knowledge/Discourse

Understanding key differences between declarative and procedural knowledge or discourse is foundational to understanding the creation of procedural discourse. On the one hand, technical specifications embody descriptive or declarative knowledge as they describe the features or architecture of a technology. Specifications writers aim to inform an audience of technical experts about the structure of a given technology or system. In contrast to specifications, user documentation embodies procedural knowledge as it teaches users how to employ software features in order to accomplish tasks. Documentation writers usually aim to teach users how to

employ technology within the contexts in which they will use it. In my understanding, procedural discourse is only a part of procedural knowledge; the knowledge exists both in the discourse and in the doing.

I found several important differences between technical specifications (see Figure 1) and end-user documentation (see Figure 2) in the examples that I collected during these two studies of technical communicators. Although both examples were written about the same software application that helped users access and provide financial services, they display different approaches because of their different purposes. (I have deleted some material from these pages for purposes of confidentiality.) The specifications in Figure 1 describe the features of the software so that technical designers can understand them. The vocabulary includes highly specialized terminology that may not be familiar to end users. The information is organized on a grid that does not provide a sense of the processes of software use. The information is presumably accurate and complete, but users would be hard-pressed to figure out how to use the software if they read only these specifications. (For additional examples and discussion of the limitations of descriptive, system-oriented documentation, see Salvo, Zoetewey, & Agena, 2007). These specifications clearly embody declarative, descriptive knowledge.

In contrast, the end-user documentation in Figure 2, an excerpt from a user manual, contains user tasks and actions, explaining only as much about the structure of the software as is necessary to teach learners to complete those tasks. This page includes not only the steps that users need to complete but also the reactions of the interface. The vocabulary is nontechnical, informing end users who are not technical experts. Step 4 indicates that nonroutine actions may be possible, a topic useful for end users who often face branching processes. From this page, users can gain a sense of the processes to be followed; they will not gain an in-depth understanding of the architecture of the software, but they probably do not need that understanding.

The technical communicators who created the user manual did not make the specifications more user-friendly merely by introducing simpler words; rather they transformed these declarative specifications by reorganizing, selecting, and reformatting content that they learned through studying the specifications and using other means of understanding the software (see Hovde, 2001; Hovde & Hovde, 2002). In addition, they considered the users, the organization, and genre conventions as they created discourse that would teach users how to work with the application. The creators of this manual accomplished tasks more complex than that of merely rewording the specifications. They created new procedural meaning—a task more

Program Specifications
Loan — Common Application Select Dialog Boxes

Data Reference and Validation
Loan Summary

Screen Literal	*Validation Criteria*	*Message Code*	*Validation Type*	*Size*	*Field Factor*	*Default Factor*	*Field Protection*
Borrower's SSN	SSN type (numeric)	Beep	Input type	9		Calling window	Field is not accessible when on the Scroll section of the screen
	Length equal 9	LAP0008	Post edit				
	Non-blank	LAP0008	Process edit				
Name Locate							
Screen Literal	*Validation Criteria*	*Message Code*	*Validation Type*	*Size*	*Field Factor*	*Default Factor*	*Field Protection*
Last Name	Alphanumeric	Beep	Input type	29		Blank	Fields are not accessible when on the Scroll section of the screen.
	Non-* only	LAP0050	Post edit				
	Non-blank	LAP0050	Process edit				
First name	Alphanumeric	Beep	Input type	20		Blank	

Figure 1. Example of Technical Specifications Describing Software Features

Chapter 2 Creating Loan Applications – Selecting an Entry Mode
→ **To view loan records by applicant name:** 1. Press <**F10**>to move the cursor to the action bar. 2. Type ***V*** to select **View**. The View pull-down menu appears. 3. Type ***N*** to select **Name locate . . .** The Name locate dialog box appears. 4. Type the applicant's last name up to 29 characters in **Last** field. You may type a partial last name followed by an asterisk (*). For example, if you wish to search for all applicants with a last name beginning with "John," type ***JOHN****. 5. Press <**F9**>. The SSNs of all borrowers with matching names appear. 6. Page through the records, using the arrow keys, until the record you want to select is highlighted. 7. Press <**Enter**>. Doing so removes the dialog box and returns you to the Entry and Edit selection window. The borrower's SSN appears in the Borrower field.

Figure 2. Example of a User-Manual Page Showing How to Use Software

challenging than merely repackaging existing declarative knowledge (see Johnson, 1998, pp. 125-129). This transformation implies that technical communicators' expertise lies in more than just "translating" or "transmitting" knowledge (Slack et al., 1993).

Perspectives on Technical Communicators as Creators of Knowledge

Some researchers have attempted to explain technical communicators' expertise, but these descriptions often do not include how technical communicators create meaning. For instance, Regli (1999) argued that technical communicators' role in creating knowledge is to employ "rhetorical heuristics to situations of interdisciplinary collaboration" (p. 31; see also Henry, 2000) and thus provide guidance to a process of shaping meaning within teams. Regli suggested that a technical communicator is not a subject matter expert (SME) and therefore may not really be seen as an "inventor" (p. 33). Instead, she concluded, if knowledge functions as an activity rather than a commodity, then a technical communicator knows how to "'perform' knowledge in a communal, dialectical context—how to orchestrate the conversation of a team of specialists working to invent, develop, produce, and test software products" (p. 34). Although technical communicators typically are not SMEs, they likely do more than just facilitate documentation processes by managing teams and shaping their discourse.

Other researchers have explored the contributions that technical communicators might provide to the meaning-making enterprises of their organizations, but these approaches do not address the possibility that technical communicators create procedural knowledge/discourse. Hughes (2002) and Applen (2002) explored how technical communicators create declarative or factual meaning within their organizations but not how they create procedural discourse for external users. In Hughes's and Applen's views, technical communicators create knowledge only by helping SMEs articulate tacit knowledge and by preserving that knowledge in forms that are durable for the organization. Hughes argued that technical communicators "negotiate meaning within development communities and between those communities and user contexts, and then capture the resulting consensus as knowledge assets" (p. 278). Although technical communicators' contributions to an organization's declarative knowledge may be important, these studies do not indicate how technical communicators shape procedural knowledge for end users.

Fortunately, some scholars have begun to understand that technical communicators as authors create more than declarative knowledge. Slack et al. (1993) have argued that "it is impossible to recognize the real power of technical discourse without understanding its role in the articulation and rearticulation of meaning . . . . This understanding would thus empower the discourse of technical communicators by recognizing their full authorial role" (p. 33). Johnson (1997) also has argued that effective user documentation should transform declarative knowledge of a system so that new knowledge can meet the goal of use within the user's context (p. 370). But even scholars who have argued for technical communicators as procedural knowledge makers have not provided a fine-grained, observationally based look at factors that technical communicators employ as they create procedural discourse.

Contextual Factors That Shape Discourse Decisions

To understand how technical communicators make meaning, we can gain useful insights by understanding the reasons behind their discourse decisions. Several composition scholars have provided categories that may be useful for understanding these reasons. These categories include considering the readers' needs and organizational constraints (Charney, Reder, & Wells, 1988) and understanding the users' situations, their goals, the technical system, and the conventions of written procedures (Farkas, 1999). A richer set of categories comes from Herrington (1985), who argued that

discourse decisions appear to consist of "audience-based, writer-based, subject-based, and text-based" reasons (p. 339). These four categories are consistent with Kinneavy's (1971) model for the aims of discourse, providing a solid, comprehensive foundation for analysis of technical communicators' decisions.

Because of the richness of Herrington's (1985) and Kinneavy's (1971) categories, I employ them in analyzing the factors I observed that influenced the shaping of discourse. I also discuss the perspectives various theorists have provided about these four factors in relation to the creation of end-user procedural discourse. In these studies, I have employed the suggestion of Paré and Smart (1994) to examine "writing processes" and "social roles to see how they might act as positive or negative heuristics" (p. 153) in the processes of discourse creation. These studies show some of the factors that guide technical communicators in creating procedural knowledge/discourse for users. In addition, these observations illuminate the expertise that shapes these technical communicators' decision-making processes within their organizational contexts. These studies explore the factors that guided these technical communicators as they transformed declarative knowledge about software and other matters into procedural knowledge in user documentation.

Research Approaches

To begin to understand the processes of shaping procedural discourse, I explored the following research questions:

- How did these technical communicators create procedural knowledge in user documentation? What factors contributed to their shaping of meaning? How useful were the categories proposed by Herrington (1985) for this analysis? What additional categories, if any, did these technical communicators employ? Which categories influenced these technical communicators the most?
- How can these findings enhance our understanding of the creation of procedural discourse?

Using data derived from observations and interviews from two case studies, I examined closely the reasons that technical communicators provided for decisions they made as they created procedural discourse. (In this article, the term *technical communicators* encompasses both technical writers, who created paper manuals and online documentation, and technical

trainers, who created training materials and led face-to-face technical training sessions.) Analyzing the reasons that guided these technical communicators' discourse decisions can provide a fine-grained picture of how creators of several forms of procedural discourse create meaning.

In the first study, I observed four technical writers in two organizations who created end-user documentation for software developed in-house. The first site in this study, a small organization that I call B&F Programming, employed four people. This organization produced and marketed software to a small, specialized user base. The B&F technical writer I studied, Sue (all names are pseudonyms), worked there in the summers and part-time during the school year. A relative newcomer to the organization, she revised paper manuals for two major software products. The process of creating documentation there was loosely structured; she was basically free to work as she pleased, but she received little support in her work process from other employees at the organization.

The second site in this study, a larger, nonprofit organization that I call Money Services, provided financial services to other nonprofit organizations. The software they sold to clients allowed them to provide these services more efficiently. The key informant at that site, Trish (a technical writer II), had about five more years of experience as a technical communicator than Sue had. Although Money Services had about 12 technical writers at the time of my data collection, I focused on three, Trish, Hannah (a senior technical writer), and Faith (a technical writer I), as they created paper and online documentation for a software package that I call QuickCash. At this site, the process for creating documentation was highly structured, with writers producing a new version every 6 months as the software was updated. These writers enjoyed generally good access to technical information and other resources that helped them complete their work on schedule. During the time of my observations there, the technical writers reorganized the overall structure of their manuals to follow the flow of user tasks, as usability experts recommend, rather than the structure of the software. This reorganization process provided a rich opportunity to observe their expressed reasons for discourse decisions based on users and the software.

I observed and recorded the interactions of these technical writers at the two sites one or two days a week for a period of 10 months. The participants also kept a notebook in which to record significant events when I was not present. I noted especially meetings and interactions during which they made decisions about user manuals with coworkers. While on site, I asked the participants to talk aloud about what they were doing as long as it did not

interfere with their work. I obtained additional data about their reasons for discourse decisions from extensive field notes, tape recordings of several meetings, collected documents, and interviews with these technical writers and their coworkers.

In the second study, three technical trainers at two university libraries produced written documentation to supplement their face-to-face training discourse. Over 2 years, I visited the sites repeatedly to observe training practices, interview key participants, collect print materials, and attend relevant meetings. At both sites, the trainers followed a largely self-directed process of creating training experiences. Although they consulted with colleagues and considered the academic calendar in scheduling training, they had a flexible process for creating training materials.

University A had a School of Library and Information Science from which graduate assistants (GAs) were hired each year to staff the reference desk and information desk in the main library. My primary informant there, Deidre, a library faculty member, trained GAs in using the technology and working within the reference area and occasionally led training sessions for library patrons. She developed a GA manual that was used during the initial training at the beginning of fall semester and then throughout the year for ongoing training. When the libraries at University A adopted a new online public-access catalog (OPAC) designed by an outside vendor, Deidre helped design and conduct the technical training of many kinds of library employees. I observed her participate in the planning for the staff and faculty training in the use of this new software. In addition, employees needed to be trained to use the libraries' CD-ROM databases.

The libraries at University B had developed an extensive employee training and development program with two full-time, in-house employee technology trainers, Tim and Yvonne, my primary informants there. Employees needed training on using the libraries' Web-based OPAC and Web-based and CD-ROM databases. In addition, University B's library administration encouraged faculty and staff members to develop Web sites for patrons, a task that required specialized training.

I heard most of the remarks that illuminated these technical communicators' reasoning during the course of general conversations and observations, but on at least one occasion, I conducted a discourse-based interview with Deidre, asking about her reasons for document design decisions in a specific manual. In conducting both studies, I took measures to minimize the impact of researcher effect although my presence no doubt slightly altered participants' behavior. Both studies met the requirements of my institutions for research with human subjects.

Although the roles and responsibilities of the technical writers at the first sites differed from those of the technical trainers at the second site, I observed several similar patterns in the reasons they provided for their discourse decisions. Instances when the types of reasons differed provided useful insights into how contextual factors could influence assumptions that guided discourse decisions.

In analyzing my findings, I initially sought in my qualitative data all instances in which these technical communicators provided a reason for a discourse decision. Once I discerned these instances, I used Herrington's (1985) categories to divide the reasons. In doing so, I discovered that some categories contained several facets, so I further divided the instances into subcategories, as I discuss in the next section. I was unable to use additional coders. The findings present a starting point for exploring how technical communicators shape meaning in procedural discourse.

Findings

The technical communicators in these studies provided reasons for their document decisions based on four general factors similar to those Herrington (1985) mentioned: understandings of users, organizational image and constraints, software structure and features, and genre conventions, as diagrammed in Figure 3. Employing these factors in discourse decisions demonstrates these technical communicators' expertise and their processes of transforming their declarative knowledge about these four factors into procedural discourse.

In discussing my findings, I present representative samples of the reasons I discerned based on my conversations with and observations of these studies' participants. Reasons in addition to the ones they expressed may have existed in these technical communicators' minds. But the fact that most of these categories of reasons were mentioned multiple times indicates that these were influential reasons for these technical communicators.

Although I discuss these categories in isolation, multiple factors frequently influenced a discourse decision. In one instance, a technical communicator needed to define the term *request* in her documentation for a new software application because she believed that users would think first of the meaning of request that existed in the old application. In this instance, both user knowledge and software knowledge motivated her discourse decision. This overlapping of influences occurred at times, but usually one category of reason dominated a decision.

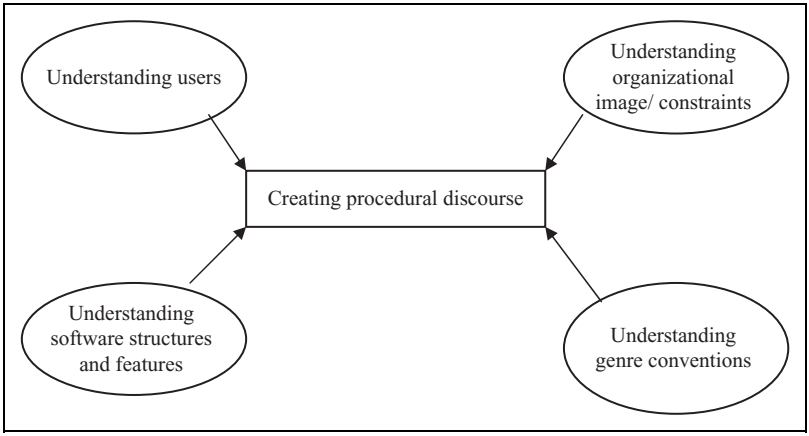

Figure 3. Four Areas of Knowledge That Technical Communicators Integrated as They Created Procedural Discourse

After providing background on each of the four factors, I discuss each in order of the frequency of occurrence in these studies. This discussion provides a detailed examination of the factors that influenced the shaping of procedural knowledge/discourse in these settings.

Decisions Guided by User or Audience Considerations

In these studies, technical communicators gave user-based reasons for their decisions in shaping procedural discourse as often (57 instances) as they mentioned the other three reasons combined. This observation is consistent with the emphasis that technical communication experts have given to how audience plays an epistemic role and provides reasons for discourse decisions in technical communication practice. Regli (1999) and Hovde (2000) acknowledged that technical communicators have expertise in analyzing and creating an image of the audience for a software application and its documentation. Technical communicators can use an internalized model of readers as a basis for discourse decisions (Wright, 1994, p. 29), and their interactions with users can provide them with insight that can be used to "argue for design changes" (Johnson, 1997, p. 370) as they "effectively implement audience knowledge . . . into the development process" (pp. 367-368). Usability specialists recognize the importance of user-centered design that includes "understanding the users, the tasks that users perform, and the environment in which users perform these tasks" (Barnum, 2002,

p. 7). Considering audience creates a "cooperative communicator," one who does not merely disseminate information but rather anticipates readers' needs and attempts to fulfill them (Wright, 1994, p. 13). In this participatory model, technical communicators "encourage readers to engage in specific kinds of reading + thinking activities. This goal is clearly different from that of imparting information" (p. 18). Considering audience helps technical communicators decide how to shape meaning in procedural discourse.

The primacy of audience also indicates technical communicators' views of the relationship of knowledge to discourse and consequently the status of their work. As Henry (2000) has noted, "If the audience is positioned merely as a consumer, the writer's role is reduced to 'low practice': adjusting appeals for maximum consumption" (p. 147). If the goal of technical communication is only to determine "the best packet and channel through which to send the information" (Johnson-Eilola, 1997, p. 69), then creating or employing a rich image of users is not necessary. If technical communicators do not see audience as epistemic, then they view information as existing outside of discourse, their readers as passive recipients, and their own function as only translators or transmitters. But if audience is a central influence in technical communicators' discourse decisions, they can more appropriately transform declarative knowledge to fit the needs of users of procedural discourse.

Creating an image of users' procedural-knowledge needs may be complicated because "users can construct working images of a technology in many ways" (Paradis, 1991, p. 263). Furthermore, according to various scholars, technical communicators may need to consider multiple facets of audience when making discourse decisions:

- Audience's interests can be central to invention, arrangement, and style decisions (Henry, 1998; Selzer, 1983) and can help technical communicators "generate a rhetorical plan" (Wells, 1986, p. 253).
- Knowledge about users' attitudes can help technical communicators assist users in feeling safe or comfortable with a new technology so that users will get started quickly and persist in learning the necessary tasks (Mardsjo, 1994, p. 192).
- Knowledge of user tasks can provide a major rationale for content decisions as technical communicators ascertain whether they have covered all major user tasks and explained the reasons or conditions associated with optional tasks (Hargis, 1998, p. 51). Technical communicators need to determine if users typically employ instructions to complete an immediate "task" or to build a generalizable "skill"

(Wright, 1994, p. 17). Technical communicators may also need to consider that workers with lower level positions are typically assigned to relatively routine and repetitive work whereas workers with higher level positions have more opportunity for nonroutine work (Johnson-Eilola, 1997, p. 228).

- Knowledge of users' experience levels with the technology and the contexts of use can help guide technical communicators' decisions (Hargis, p. 56).
- Understanding users' expectations for the conventions of instructional discourse can aid technical communicators because "patterns help users become comfortable with the organization of information and focus on the content they need" (Hargis, p. 62).
- General knowledge of how users learn can help technical communicators decide on their level of elaboration (Charney et al., 1988).
- Knowledge of user contexts can help technical communicators "interpret product functionality in light of both the user contexts and the developers' intentions" (Hughes, 2002, p. 277; see also Suchman, 1987).

Thus, technical communicators' knowledge of multiple facets of audience guides their discourse decisions (Porter, 1992, pp. 130-131); these complexities contribute to the challenges of maintaining an appropriate image of users while creating procedural discourse.

Before technical communicators can employ their knowledge about users, they need to have access to information about those users. When professional communicators have an inadequate image of users, they find their work more challenging (Hovde, 2000; Smart, 1999a), and advice to "include enough detail for users to perform their tasks, but no more detail than they need" (Hargis, 1998, p. 55) or to make "the user's ignorance explicit" (Hughes, 2002, p. 279) or to put yourself in the reader's shoes (Hargis, 1998, p. 55) is not helpful.

The technical communicators in this study enjoyed varying levels of direct and indirect access to users, conditions that influenced how rich an image they could create and use. Sue at B&F had limited contact with end users; she sometimes provided help over the phone to users who called, and she met with users face-to-face about once a year.

Trish, Faith, and Hannah at Money Services had even less access to external users, but they had better access to internal readers (software developers), as in Raven's (1992) study. Therefore, these technical writers used

their mental constructs of the internal readers to shape manuals so that they would be understandable to the systems developers. (These developers, who appreciated the manuals more than they did the technical specifications that other technical people developed, frequently consulted the technical writers' manuals rather than the specifications in order to become familiar with a software module that another team had developed.) Technical writers at Money Services encountered difficulty in moving from writing such system-oriented documentation to writing the new documentation that was oriented toward user tasks largely because they did not understand external users well. Because Money Services offered no built-in mechanisms for external-user observations or usability testing of their documentation, the technical writers struggled especially when deciding on the transitions and the "flow" as they reorganized the documentation (for more information about how these technical communicators learned about users, see Hovde, 2000).

In contrast, Deidre, Tim, and Yvonne, as technical trainers, enjoyed much greater face-to-face access to users, especially during training sessions, and were thus able to shape training materials and experiences to meet user needs (for more insights, see Hovde & Hovde, 2002). At all four sites of my two studies, little formal analysis of the users was done, especially at the first two sites. Rather, these technical writers gradually built images of their users mostly by indirect means. The library technical trainers, however, through direct interactions, probably had the richest image of their users. All these technical communicators created procedural discourse for many kinds of users, and so they considered many facets of those users, even if they did not have a rich image of them developed through extensive interactions. Even those technical communicators who had limited access to users still provided many user-based reasons for decisions, indicating the strength of the factor of understanding users.

As I analyzed my observations, I considered the following questions based on my review of the literature: Did these technical communicators employ their understanding of users' interests, attitudes, tasks, experience levels, genre expectations, learning styles, and contexts to guide their decisions on invention, arrangement, and style as the literature suggests? If so, how? What other types of reasons might these technical communicators provide? After discussing their reasons, I compare the reasons with the list generated from my review of the literature.

In 57 separate instances, these technical communicators mentioned audience-related reasons for decisions they made while creating end-user manuals or training materials. As I analyzed my data, seven categories of

audience-based reasons emerged: users' routine and nonroutine uses of the software, users' expressed desires, users' needs for declarative software knowledge, users' needs for nonsoftware knowledge, users' learning styles, users' desires for efficient processes, and users' contexts of use. I discuss each category in order of frequency.

Users' Routine and Nonroutine Uses of the Software. The most common audience-based reason for discourse decisions came from these technical communicators' understandings of how users employed the software in both routine and nonroutine situations (see Table 1 for selected examples that illustrate these decisions). This understanding of common tasks allowed technical communicators to select topics and design approaches in order to teach users. These technical communicators, especially those in the libraries who were able to form rich images of their users through direct contact, provided 21 reasons (of the 57) that demonstrated their understanding of how users employed software. Consistent with what Mardsjo (1994) has noted, they needed to select procedures and shape discourse to aid users but not overwhelm them with unneeded declarative information about the software. This selection often proved difficult, especially at Money Services, where the internal readers who were also software developers exerted a great deal of influence on the technical communicators' documentation. Nevertheless, having a sense of how users might interact with the software enabled these technical communicators to transform declarative knowledge into procedural discourse. These technical communicators appear to have been aware that end users primarily needed procedural, not declarative, knowledge consistent with what Hargis (1998), Wright (1994), and Johnson-Eilola (1997) have argued.

Users' Expressed Desires. When these technical communicators had access to users' expressed desires, they used these understandings as a basis for discourse decisions (see Table 2). The more direct access to users that these technical communicators had, the richer their images of users became, and these images guided discourse decisions. The examples in this category show the technical communicators as responsive to users' requests and suggestions, using an interactive approach to create meaning. Although some of these reasons may overlap slightly with some of the other categories, I separate them into their own category because they were based on direct requests from users rather than on the technical communicators' indirect extrapolation of user images as in some of the other categories. These technical communicators, especially the trainers, provided 10 instances in which users' expressed preferences and desires guided their discourse

Table 1. Discourse Decisions Based on Users' Routine and Nonroutine Uses of the Software

Decision	Reason
Providing standardized formats, organizing material by keystrokes, and creating more sections in revised manuals	To help users locate information quickly
Using bold text instead of quotation marks to indicate material to be typed	To prevent users from typing the quotation marks as well as the relevant text
Repeating information in various volumes of the set of manuals	To assist users who might not use all the manuals or modules of the software
Indexing the new version of the manual by common terms rather than headings, as had been done in the old manuals	To provide familiar language that would aid the users' searches
Organizing the manuals according to user tasks, not software parts as had been employed in the past	To guide users in completing common tasks
Designing training materials to be used as a guide during training and as a reference document later	To provide for both categories of use of the materials
Adding information in the manuals about how to add and edit performance information	To help users who were having difficulty with these tasks, according to reports from user contacts
Explaining branching processes by separating them into parallel tracks with different outcomes	To help users follow nonroutine tasks
Placing supplemental information in sidebars and call-out boxes in the manuals	To show that these processes were outside the normal flow of tasks
Providing trainees with declarative knowledge and with opportunities to try new tasks during training sessions	To assist trainees with completing nonroutine tasks after the training sessions
Providing training only on the functions of the software that users were likely to employ	To provide for the needs of trainees who only needed fluency in the software, not mastery

decisions. Many of these user requests did not relate directly to procedural knowledge but rather to the means by which procedural knowledge was communicated (e.g., requests for what topics to include in manuals),

Table 2. Discourse Decisions Based on Users' Expressed Desires

Decision	Reason
Dividing a manual into sections with tabs and divider pages	Users had asked for such a format.
Including a list of pros and cons about new ways of completing tasks	Trainees had expressed annoyance at the newest interface for specific software application.
Including additional information in a new version of the manual about how to make judgments	Users had asked for judgment information.
Developing advanced training workshops	Trainees had expressed annoyance at attending workshops that were too basic.

indicating that technical communicators need to be aware of the communication means that engage users.

Users' Needs for Declarative Software Knowledge. Although these technical communicators intended to teach primarily procedural knowledge, they recognized that users sometimes need declarative software knowledge in order to complete nonroutine or unanticipated tasks. At other times, however, these technical communicators elected to exclude irrelevant facts about the software because they believed that users would not need those facts in order to develop their own procedural knowledge (see examples in Table 3). This finding is consonant with Mardsjo's (1994) contention that declarative technical content should be selected according to users' needs. The fact that users' declarative knowledge needs was the third-most common reason expressed (seven instances) shows that these technical communicators understood that users need both procedural and declarative knowledge in procedural discourse. These examples indicate that these technical communicators needed to understand the features of the software and how users might employ declarative knowledge as they use the software. As Karreman (2004) has indicated, an appropriate level of declarative knowledge may help users to retain procedural knowledge and to deal with nonroutine situations. The declarative discourse that these technical communicators selected, however, generally did not overwhelm the texts they produced, indicating that they understood that procedural discourse provided greater utility to users.

Table 3. Discourse Decisions Based on Users' Needs for Declarative Software Knowledge

Decision	Reason
Showing only the speed keys and not all the menus and cascades in the manuals	Users would probably not need that information.
Excluding basic software information from the manuals	Marketing representatives had already shown users the basics of the software.
Including information about only the most important error messages	Users would not need to know about less important error messages.
Asking trainees why they thought software behaved in a specific way	Trainees would develop an understanding of the principles behind the software design.

Table 4. Discourse Decisions Based on Users' Needs for Nonsoftware Knowledge

Decision	Reason
Excluding definitions of accounting terminology in the manual for accounting software	The technical writer assumed that users knew the terms and would be insulted if the technical writer defined them.
Including a data analysis guide in the manuals	Users would need these guides to help interpret reports generated by the software.
Defining the term *database control number* in the manual	Technical writers noted that users did not know what the term meant.

Users' Needs for Nonsoftware Knowledge. These technical communicators' perceptions of users' general knowledge outside of the software (see examples in Table 4) guided their discourse decision making as frequently (seven instances) in these studies as did their perceptions of users' needs for declarative knowledge. Reasons in this category deal mainly with users' need for declarative, nonsoftware knowledge that is essential for their employment of procedural knowledge. The presence of this reason indicates that these technical communicators possessed some knowledge of users' general knowledge while they were focusing on teaching them how to use the software. Nevertheless, the users' need for nonsoftware contextual knowledge guided as many of the discourse decisions as did the users' needs for declarative software knowledge.

Table 5. Discourse Decisions Based on Users' Learning Styles

Decision	Reason
Including only the most complicated tasks in the take-home training materials	The trainees had mastered the more basic material in the training session.
Offering training sessions in increments over time	The trainees could practice and absorb procedural knowledge over time.

Users' Learning Styles. Consistent with the assertion of Charney et al. (1988) that technical communicators need to know how users learn, these technical communicators based several decisions on their understandings of how users become skilled at using the software. These technical communicators especially focused on drawing analogies between users' given knowledge and their new knowledge (see examples in Table 5). Although the technical writers never mentioned this reason, the three trainers provided this reason six times during my observations, indicating that close interaction with users allowed the trainers to consider how users learned and completed tasks. These trainers showed that they understood that users' procedural knowledge could not develop quickly. Their focus on these analogies and their ordering and pacing of the instructions indicated that the technical trainers realized that their users were engaged in a process of learning, both during and after the training sessions. In expressing these reasons, trainers indicated not only that they understood users' procedural knowledge needs but also that they understood that learning itself was a process that deserved care and attention. The fact that the technical writers did not mention this reason indicates that they may have focused less on the users' process of learning than they did on other audience features.

Users' Desires for Efficient Processes. These technical communicators demonstrated their understanding of users' desires for efficient and effective processes when using the software, mentioning this audience-based reason four times. Because of this understanding, the technical communicators reminded users to back up their files daily, explained how to search databases efficiently, showed how to display results using little space to save on printing costs, and discussed how to evaluate the results of a search. Although the technical communicators did not frequently offer users' desires for efficiency as a reason, possibly because of their limited knowledge of users' contexts, by suggesting efficient strategies, they

indicated that they understood at least a portion of the users' contexts in employing procedural knowledge.

Users' Contexts of Use. Consonant with Hughes's (2002) argument about the need for understanding users' contexts, these technical communicators in two instances gave reasons based on their understanding of users' situations. One technical writer at Money Services realized that some of her users were working on Macintoshes, so she created a separate "getting started" manual for them. In another instance, a technical trainer provided instructions on the differences between accessing software on a workstation in the library and accessing it from a home computer. Ignoring the contexts in which users work can lead to less effective user documentation. But these technical communicators rarely mentioned this reason, probably due to the inadequate image that some of these technical communicators had of users' contexts. Tellingly, these technical communicators often expressed a desire for a greater understanding of users' contexts so that they could shape procedural discourse more appropriately.

Summary of Audience Considerations. These technical communicators mentioned only four of the audience considerations covered in the literature although they cited four additional considerations as reasons for their decisions while creating procedural discourse (see Table 6). The category of users' tasks was the most frequently mentioned reason, indicating a strong agreement with the literature. The other two factors that overlapped with the literature were among the least frequently mentioned. Three of the four most frequently mentioned factors did not appear in the literature, indicating that my studies offer a deeper understanding of some of the complex audience considerations that guide technical communicators' shaping of procedural knowledge.

These technical communicators overall most frequently provided audience considerations (57 in all) as reasons for their decisions (in contrast to the other three factors that I discuss next), indicating that a rich picture of users strongly influenced their discourse decisions (as in Wright, 1994). Even at sites with limited access to users, the technical communicators employed their relatively poor images of the users to guide their discourse decisions. These technical communicators needed to understand users' potential processes in using the software, but they also needed to consider their needs for declarative knowledge. A sense of users' learning styles and contexts also guided these technical communicators' decisions

Table 6. Summary of Technical Communicators' Audience Considerations Compared to Those Cited in the Literature

Types of User Knowledge	Considerations Found in the Literature	Considerations Found in This Study (Number of Instances)
Users' tasks, routine and nonroutine	X	X (21)
Users' expressed desires		X (10)
Users' declarative knowledge needs about the software		X (7)
Users' nonsoftware knowledge needs		X (7)
Users' learning styles	X	X (6)
Users' desires for efficient strategies		X (4)
Users' contexts of use	X	X (2)
Users' interests	X	
Users' attitudes	X	
Users' experience levels	X	
Users' expectations of conventions	X	

as did their close attention to users' expressed wishes and nonsoftware knowledge needs.

Based on these observations, I argue that technical communicators' procedural discourse may profit most from gaining knowledge about users. The richness and completeness of the audience image that technical communicators held strongly influenced the appropriateness of their communication decisions, but the social contexts in which these technical communicators worked also strongly influenced their decisions.

Decisions Guided by Organizational Image and Constraints

After audience concerns, the second-most frequently mentioned factor (21 instances) that guided decisions was the image and constraints of the organizations in which these technical communicators worked. Although Herrington (1985) mentioned "writer-based" reasons, in an organization, an individual is often not the sole author of any one piece of communication. Therefore, I have expanded this category to include the organizational image and constraints as a guiding factor.

In general, as technical communicators make discourse decisions, they consider the image that the organization wants to portray because they communicate as the "voice" of an organization (Mathes & Stevenson, 1991, p. 5), not as autonomous individuals. In fact, Henry (2000) suggested that "in some organizational contexts, organizational goals and organizational authorship are nearly synonymous, so much does the organization exist through and by virtue of its representation to a specific clientele through specific documents" (p. 88). Technical communicators need to understand and respond appropriately to organizational conditions as they create procedural discourse (Hovde, 2002). Because discourse is created in social contexts, factors within the community either aid or hinder the creation of knowledge in discourse. Thus, we can take "meaning production out of the heads of individual speakers and locate it in the fields of social interaction" (Hanks, 1991, p. 13). Technical communicators who work within a *community of practice*, defined as "a set of relations among persons, activity, and world, over time and in relation with other tangential and overlapping communities of practice" (Lave & Wenger, 1991, p. 98), function within a community in which shared understandings of meaning are created via communication.

Organizations influence technical communicators throughout all phases of writing. Even before technical communicators begin drafting, they organize, select, and connect what they already know with the organizationally generated goals of the document (Spivey, 1997, p. 136). A technical communicator's socially influenced "mental product" or internal representation of the goals of the organization affect the creation of their "textual product" (p. 124). Technical communicators are not isolated inventors who determine their own goals and choose how they make meaning. Organizational power dynamics often contribute to technical communicators' beliefs that their work is considered unimportant or is misunderstood. In some organizational environments, "writers' second-class, support-staff, functionary status inevitably undercuts their effectiveness." People in many organizations think of "form as separate from content and writing as a mode for communicating predetermined thought rather than for exploring and instantiating reality" (Henry, 2000, p. 88), thereby excluding technical communicators from useful resources that the organization might provide in order to produce user documentation that would benefit both the end users and the organization's image (p. 180). These resources might include access to users, to usability testing processes, or to information about the software (Hovde, 2002). As they make discourse decisions, technical communicators deal with the interests of the organization—while attempting to

serve users who need procedural knowledge. Naturally, these interests may conflict; at times, the expectations of the organization trump other reasons for discourse decisions (Mardsjo, 1994, p. 191). Technical communicators may not be able to make high-level discourse decisions on their own, even if they have knowledge that could improve the user documentation.

In considering the organization's culture, technical communicators gain a broader perspective of the goals influencing their discourse. Within the four organizations in this study, technical communicators' status ranged from very low to moderately influential. At B&F, the lone technical writer's status could best be described as marginal. When she returned from her summer internship to her university studies in the fall, the president of the company did not intend to replace her; she was given minimal resources with which to complete her work. At Money Services, technical communicators were more highly valued than at B&F, as demonstrated by several people from other departments expressing how much they admired and appreciated the work that the technical writers did. Management supported their work by requiring programmers to review documentation in a timely manner and to write specifications for every feature of all the modules. Paradoxically, at that same organization, technical writers were also devalued by their remote location from the main office, their late participation in the software development cycle, and their limited access to external users. The situations were better at the two academic sites, with most of the technical trainers having sufficient independence and adequate resources to conduct their work efficiently. They were well appreciated at both sites. These organizational conditions and constraints directly and indirectly influenced these technical communicators' decisions about discourse.

The only instance I observed in which organizational image or ethos directly motivated a discourse decision occurred when someone outside of the Technical Writing Department at Money Services expressed the desire that the documentation look "professional." Consequently, the organization dedicated more resources to achieving that goal, and the technical writers participated willingly. Although organizational image was mentioned only once as a reason for a discourse decision, I did discern that constraints such as organizational opinions, needs, and general conditions were mentioned 20 times as reasons for discourse decisions, far fewer than audience reasons but more than content or genre reasons.

Because organizational conditions influenced discourse decisions in publications disseminated to external users, the conditions within B&F and Money Services led to discourse that reflected the ethos of those

Table 7. Discourse Decisions Based on Communication Opinions of Coworkers

Discourse Decision	Reason Based on Coworkers' Communication Opinion
The technical writer designed the manual in 2-page modules even though the information was crowded onto the pages	B&F's president wanted the manual organized in modules that fit on two facing pages
The technical writer returned to a previous design for the manual pages	The president and salesperson-trainer did not like the technical writer's redesign of some of the pages
The technical writer organized the manual by the chronology of typical tasks	The president approved of the technical writer's suggestion to reorganize the manual
The technical writer revised manuals to be simpler	The salesperson-trainer reviewed manuals before publication and requested that they be designed more simply
Technical writers included a chart in the front pocket of the manual showing all the pull-down menus for what was covered in that manual	People in client support wanted an overview for users of the documentation
Technical writers created a smaller quick reference manual to accompany the large manuals for one module	A person in User Services did not think that large manuals met users' needs

organizations. In the library sites where technical communicators trained only internal staff, I found that these technical communicators were less concerned about organizational constraints and image. Also, because the libraries were not concerned with profits, costs were not as much of a constraint there as they were in the corporate settings. Consistent with Mardsjo (1994), organizational influences sometimes overrode the standards that these technical communicators held for creating effective user documentation.

The organizational influences I observed fell roughly into three categories. First, the communication opinions of coworkers (some of them not based on sound communication principles) guided some discourse decisions (see examples in Table 7). When these opinions came from influential people, the technical communicators were more likely to acquiesce, even when they believed that these decisions based on those opinions made the documentation weaker. Second, the documentation-related requests of

Table 8. Discourse Decisions Based on Coworkers' Requests

Discourse Decision	Reason Based on Coworkers' Request
The technical writer planned to create a manual for a specialized group of users	The salesperson-trainer at B&F requested a manual for the group
The technical writer removed one department's phone number from the list of contacts in the manual	This department did not want clients calling them with questions

others within the organization affected these technical communicators' decisions (see examples in Table 8). Although these other individuals were not end users, their proximity or access to the writers influenced the documentation. Finally, general organizational conditions, including schedules and cultural attitudes (see examples in Table 9) influenced these technical communicators' discourse decisions. At times, these conditions acted as constraints, preventing the technical communicators from developing documentation that would be maximally useful to users, but at other times, these conditions provided support to the technical communicators in achieving their goals.

This finding that technical communicators' decisions were influenced by organizational constraints is not surprising to those familiar with organizational dynamics. These technical communicators were sometimes faced with choosing between organizational values and what they believed to be effective communication. For instance, although the Money Services' manuals were seen as a reflection on the company and the software product, one manager also wanted to save money and did so at one point by downgrading the quality of the cover and tabs of the large manuals. The technical writers were not happy about this change but were required to implement it. In another instance, an organizational condition proved favorable to the creation of procedural discourse when influential individuals within Money Services gave the technical writers permission to reorganize the manuals which had been organized by the structure of the software menus, so that the new manuals would follow the typical flow of user tasks. Thus, in the contexts of varying organizational conditions, these technical communicators created procedural discourse, recognizing that ideal documentation could not always be realized.

Many of these organizationally influenced reasons did not relate directly to procedural discourse although the conditions frequently influenced the

Table 9. Discourse Decisions Based on Organizational Conditions

Discourse Decision	Reason Based on Organizational Practice or Condition
The technical writer gave high priority to requests made in a log of user phone calls when she was revising documentation	The office manager shared the log with the technical writer
The technical writer delayed writing one new manual	The president at B&F was too busy to tell the writer how he wanted the manual to look
Plans were made for quick reference manuals for all modules for external users. Simpler manuals were planned for in-house distribution only. Technical writers reduced the numbers of people receiving the large manuals and redesigned the large manuals' covers and tabs to be less expensive, even though the writers hated how the tabs looked	Upper administrators had pointed out that smaller, less elaborate manuals would save money for the company
Rather than continuous pagination, the technical writers gave each section of the new manual its own number and pagination so that they could update and send only sections of the manual that reflected changes in the software.	The software at Money Services was updated every six months.
A technical writer resisted including too much information in the documentation about nonsoftware related business procedures	Organizational culture typically viewed the documentation as supporting the software at Money Services
Trainers offered one-on-one training or made arrangements for computer-based training modules at unusual times	New library staff members were hired in the middle of the year and thus missed the group training
Training sessions were scheduled continually for the library student workers	The library's student workers frequently turned over
A trainer photocopied the documentation supplied by the vendor and included it in a section of her training manual rather than designing her own materials	The library trainer faced time limitations in the preparation of the material

context for creating that discourse. Faith, the novice technical writer at Money Services, in particular had a strong vision of how procedural discourse should be structured for users and frequently expressed frustration that organizational conditions prevented her and the other technical writers from creating procedural discourse that was more likely to be accessible to external users (i.e., organized by user tasks, indexed by topics rather than headings and subheadings). These technical communicators, however, often found ways to work around the organizational constraints in order to produce discourse that was as helpful as possible to end users. In addition, they became skillful at using the resources that the organization provided them to aid in the creation of procedural discourse.

These observations confirm that "professional writers are always balancing audience response with organizational concerns and priorities" (Henry, 2000, p. 147). Organizational constraints and resources is an area that is not explicitly included in Herrington's four factors that guide discourse decisions, but it played an influential role in these technical communicators' decisions, second in frequency only to considerations of audience and more frequent than subject-matter reasons.

Decisions Guided by Understandings of the Technology

Technical communicators need to form a declarative knowledge "profile" or "image" of the structure of the technology (Price & Korman, 1993, p. 54; see also Hovde, 2001) and then transform it into procedural discourse as they create user manuals. Given that technical communicators often possess an in-depth understanding of the technology, deciding on the level of technical information that users require in procedural discourse presents a challenge. These technical communicators selected information based on their perceptions of users' needs, a task that is complex because a manual must be accurate but should omit certain "facts" so that users will not be overwhelmed with information that is irrelevant to their tasks (Johnson-Eilola, 1997, pp. 52-53; Mardsjo, 1994, p. 188). Ideally, technical communicators can integrate their understanding of audience needs with their understanding of the software as they shape procedural discourse.

Based on this thinking, I had the following questions about content decisions: How do technical communicators employ their knowledge of subject matter when making discourse decisions? How do they choose declarative and procedural content appropriately? How might technical communicators provide accuracy without overwhelming users with more information than they need? How does declarative specification knowledge become

transformed into procedural discourse? How does technical communicators' knowledge of the software affect discourse decisions? In the following discussion of my observations at the four research sites, I begin to address some of these questions.

Technical writers at Money Services enjoyed good access to technical specifications, attended development meetings at which they received and discussed the most recent changes in specifications, and regularly received technical reviews of manuals from the software developers. Five programming teams developed five modules of the software for which the developers wrote specifications. Because the technical writers wrote user documentation for all five modules, they held a broader understanding of the complete software application than did the software developers although the developers held a deeper understanding of their respective modules.

In contrast to Money Services, the three other sites in these studies provided the technical communicators with little or no access to technical specifications or to software developers. At B&F, the president (who was also the programmer) never wrote down the revisions he made to the software, and he dedicated little time to answering Sue's questions, so she had to experiment with new versions of the software in order to try to learn its functions. At the university library sites, external vendors created the software, so the technical trainers learned about it by experimenting, reading the available documentation, attending trade shows and marketing demonstrations, and occasionally attending off-site training sessions offered by vendors. Many of these approaches left them with limited understanding of the software. (Even the technical writers at Money Services who had good access to specifications found that those specifications were not always complete or accurate.) One technical trainer expressed awareness, however, that technical knowledge was important to her work; she noted that the more familiar she was with the software, the more confident she was during the training session and the more smoothly it went.

In these studies, I observed 16 instances when these technical communicators mentioned their understanding of the software as a reason for their technical communication decisions (see examples in Table 10). Along with their knowledge of the software, these technical communicators took into account the analogies and expectations that users brought with them as they learned to use the software, including concepts users gained from having used other applications. For instance, in a new library database interface, the range of dates used in a search was preset. One technical trainer noticed that users were unhappy because they were accustomed to being able to

Table 10. Discourse Decisions Based on Understanding of the Software

Discourse Decision	Reason Based on Understanding of the Software
Writers decided to exclude examples about items that might change	Portions of the software would change in newer versions
Writers explained specialized terms	Software menus contained esoteric terms
Writers created a chart for the user manual to show when certain elements of the software might appear, even though not all of those elements appeared on the screen at one time	The software's architecture was not obvious from looking at the interface
Writers decided to create only a supplemental book for Macintosh users, despite a request from a user for an entirely separate manual	There were not many differences between how the software functioned on Macintoshes and PCs
A technical writer organized the manual by software menus although she would have preferred to do it "chronologically" according to the user's work flow	The typical chronology of use for a particular application was not clear from only looking at the interface
A technical writer followed the order of items in the interface as she created a manual	The features' order was set
A technical writer consulted the specifications but used her own knowledge of the software to fill in the information missing from the specifications	The technical specifications described most of the structure of the software
Technical writers told users what they had done in order to get a specific error message. (In previous versions of the documentation, the technical writers only repeated what the message on the screen said.)	Error messages as written by the programmers were determined not to be clear
A technical writer used the word *can* for situations in which users could only enter specific information. She reserved *may* for situations in which entering data was optional	The software limited what users could enter into a field

(continued)

Table 10 (continued)

Discourse Decision	Reason Based on Understanding of the Software
The technical writer explained one error message only generically	Two paths through the software called up the same error message
Trainers created workshops teaching users about several of the unique features of the software	Trainers noticed "eccentricities" in a new application
A trainer mentioned similarities between a familiar application and a new one	The new application was similar to other applications that users knew

create their own ranges of dates when searching databases. This trainer therefore adapted the training to account for the users' expectations from their use of previous applications.

These technical communicators also had to select topics to exclude based on their understandings of users and software. For instance, Trish at Money Services resisted discussing nonsoftware activities in the user manuals because she maintained that the writers' only responsibility was to explain the software. In other instances, technical writers included only the shortest way to complete a task rather than listing the multiple options that the software allowed. One trainer taught only the functions of the software that users were likely to need; she did not provide an entire picture of the software. She realized that not all users needed to master the software; some just needed to be at a fluent level with it. In shaping this procedural discourse, these technical communicators integrated their software knowledge with their understanding of users' needs for procedural and declarative knowledge, illustrating the interplay of procedural and declarative knowledge in creating procedural discourse.

Although technical communication on the surface appears to be mainly about technology, these technical communicators expressed fewer subject-matter reasons than audience or organizational reasons for their discourse decisions. One cause for this imbalance may be that much of these communicators' software knowledge was tacit. Another cause may be that audience and organizational considerations influenced their decisions more than did software. Nevertheless, knowledge of the software and how to select content appropriately for users was necessary so these technical communicators could transform declarative technical knowledge into procedural discourse.

Decisions Guided by Understandings of the Genre Conventions of Procedural Discourse

It may be both a blessing and a curse that technical communicators bring to collaborative situations expertise in the genre conventions of procedural discourse. On one hand, technical communicators may hold low status if coworkers believe that a communicator's expertise has to do only with forms of communication (Henry, 2000, pp. 74-75). If people see "language and discourse as mere packaging for thought" (p. 115), language work does not make meaning or add value. On the other hand, technical communicators' expertise in the conventions may help them make decisions about procedural discourse more efficiently than if they did not have a sense of the conventions. Knowing genre conventions can spare technical communicators from having to invent new communication practices and aid them in communicating with audiences who hold similar expectations.

Genre has recently taken on an expanded meaning to include the typical practices within an activity system. Because "genres in the professions serve as structures of activity and discourse for generating and applying specialized knowledges" (Smart, 1999b, p. 251), technical communicators need an awareness of the communication expectations within their activity systems, especially when these activity systems extend beyond their immediate organizational contexts. These genre conventions can be epistemic because "an organization's system of written genres can be seen as a multifaceted rhetorical structure for creating and distributing knowledge needed for carrying out the organization's work" (p. 252). I focus here on the conventions of procedural discourse genres, especially as they were understood by these technical communicators.

Typical patterns for structuring information may be "useful for ensuring that all standard parts of parallel, or similar, topics are included" (Hargis, 1998, p. 67). Paradoxically, these "existing genres and patterns for stylistic elements are strong, which leaves relatively little room for developing the text according to the situation" (Mardsjo, 1994, p. 191). As technical communicators shape procedural discourse, they must use the expected communication conventions so that the discourse will be received well by users, but the technical communicators also need to know when to break with those conventions in order to achieve communicative aims.

Within procedural discourse, conventions have emerged and continue to emerge. Hargis's (1998) checklist for completeness of technical communication artifacts focuses mainly on conventions, but these conventions employ users as a rationale for their existence (p. 71), indicating that these

conventions are not arbitrary but part of a larger system. Technical communicators need to become familiar with these conventions and know their appropriate uses. But uncritical or inappropriate use of conventions can make discourse less effective when that use does not fit a given situation. In making discourse decisions, technical communicators may use conventions to guide them in using specific forms, but they may also take into account readers' expectations within their activity systems.

Questions about the use of conventions include these: How strongly do conventions influence the discourse decisions of technical communicators, especially in comparison to the other three factors? If technical communicators are more than scribes whose expertise lies only in conventions, how do common discourse expectations influence technical communicators as they create procedural discourse? What are possible sources of conventions? I address several of these questions here.

Although we might expect that conventions would prevail over other reasons for making discourse decisions, given the relative infrequency in these studies of reasons based on conventions, these technical communicators appear to be guided less by conventions than by the other three factors. In 11 instances that I observed, the technical communicators' individual standards, the organization's standards, or external standards (see examples in Table 11) guided these technical communicators' discourse decisions. Although some of the reasons seemed to be held only by an individual, these technical communicators developed some of their standards from interaction with communication artifacts in various settings.

Despite the perception in some circles that technical communicators function as little more than scribes and that their expertise lies in their knowledge of discourse conventions, participants in this study did not mention reasons of genre conventions nearly as frequently as they mentioned users, organizational conditions, or the software as reasons for their discourse decisions. Their limited mention of conventions may indicate that these technical communicators held these understandings tacitly or that other factors were more important influences in creating procedural discourse. Although these technical communicators were sensitive to following the expected procedural discourse conventions, they also often adjusted to audience needs, especially when conventions did not give guidance.

Nearly half of the reasons based on conventions related to procedural knowledge, indicating the participants' desire to design written materials to facilitate users' ability to interpret the procedural discourse easily and efficiently. These technical communicators did not employ conventions in an arbitrary or isolated manner but integrated them with the goal of

Table 11. Discourse Decisions Based on Understanding of Conventions

Discourse Decision	Reason Based on Convention
Individual conventions	
A technical writer chose to make the manual a workbook rather than strictly "informational."	The purpose of user documentation is to be interactive
A technical writer placed supplemental information in separate text boxes because it was "outside the logical flow."	Material that is supplemental should not look like essential material
A technical writer tried not to have index entries with only one subcategory	She thought index items with only one category looked "awkward."
Organizational conventions	
A trainer decided to organize the manual in the same manner in which training was carried out	Training was typically carried out from basic to advanced topics
The technical writer placed sample reports in appendixes because there was not enough room for them on the relevant module pages even though the samples would have been more useful to users near the relevant steps	B&F's president wanted each module to cover only two facing pages
The lettering used for the name of the software was changed from "Quick-Cash" to "QUICK CASH."	Research from Money Services' legal department indicated that the software's name had been copyrighted in all capital letters
The writers changed periods to appear within quotation marks rather than after them	The organization's style guide said that periods should appear within quotation marks
External conventions	
A technical writer italicized information about the computer's responses in the manuals she was creating	Another company's manuals showed the computer's response in italics
The trainers incorporated more graphics into the training handouts	The library trainers noticed that software manuals from the vendors included many graphics

communicating procedural knowledge to users. In doing so, they also took seriously the organization's genre expectations, some of which did not necessarily relate to procedural communication.

Implications of the Findings

These findings have implications for understanding the creation of procedural discourse, for technical communication practice and instruction, and for future research.

Implications for Understanding the Creation of Procedural Discourse

My observations in these studies provide an in-depth look at factors that contribute to technical communicators' creation of procedural discourse and knowledge. These technical communicators, influenced by four factors, transformed declarative software knowledge into procedural discourse. The findings reinforce Herrington's (1985) insights about the reasons behind writers' discourse decisions. But my findings expand on one factor, that of authorial image, in discourse decisions. Herrington's work refers to authors' concern about their own image, but in my studies, the image of the organization seemed to matter more. And organizational constraints and resources appeared to play a much stronger role in this workplace communication than did organizational image. (Herrington's categories were shaped by studies of academic discourse, so it is not surprising that organizational conditions were not discerned.) These factors appear to serve well as a tool in understanding how these technical communicators made decisions in context.

These technical communicators transformed declarative knowledge about users, the organization, the software, and genre conventions into procedural knowledge for the benefit of users, with users being the strongest factor (see Figure 4). Had genre conventions been the strongest influence, we might conclude that these technical communicators were not creating meaning but merely repackaging it. But the overwhelming proportion of user, organizational, and subject-matter considerations supports the argument that technical communicators function as more than "scribes" who "'write up' technical information" (Johnson, 1997, p. 367). Even when these technical communicators based their decisions on the subject matter, they transformed declarative knowledge into procedural knowledge for users.

A rich image of users and the types of knowledge they need can especially assist technical communicators in creating discourse that will "establish perspectives in simplified points of view that clarify the structure and purpose of artifacts, yet also hide their deeper complexities" (Paradis, 1991, p. 264) so that users are not overwhelmed with more knowledge than

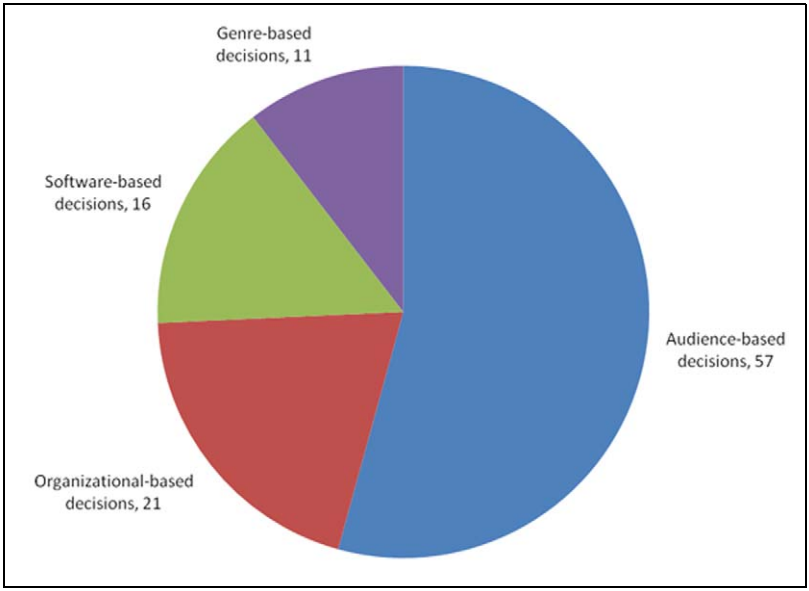

Figure 4. Relative Proportions of Each Major Category of Reasons for Procedural Discourse Decisions

they might need. These technical communicators needed to understand users' needs for productive, practical, and theoretical knowledge (Atwill, 1998) to assist them in creating effective documentation. In addition, they needed to understand users' contexts for employing the software, learning styles, typical tasks, needs for declarative knowledge, expressed preferences, desires for efficient strategies, and nonsoftware knowledge needs. If technical communicators have adequate, direct access to users and knowledge about users throughout the process, they will be more likely to create useful procedural discourse. By drawing on audience insights, technical communicators can create important connections between users and technology.

In integrating knowledge from all of these four areas, technical communicators' tasks encompassed much more than simply "writing up" information or translating technical specifications into friendlier language or transmitting information to users. Rather, these technical communicators functioned as an "intermediary between the technology producer and the technology user" (Mardsjo, 1994, p. 190), contributing to the creation and dissemination of procedural knowledge. Because they were creating meaning in complex contexts and making decisions based on many factors, they

were functioning as authors, not mere scribes. In their discourse decisions, and according to their articulated reasons for these decisions, these technical communicators moved beyond "translation" and "transmission."

Although these studies did not analyze user reactions to the documentation, when effective procedural discourse is available, users are more likely to be able to use sophisticated technology more easily. Using their expertise as authors, these technical communicators created procedural discourse with the primary aim of teaching people to use technology, an aim not covered in Kinneavy's model (1971), but one worthy of continued consideration.

Implications for Technical Communication Practice and Instruction

Practicing technical communicators may use the findings from these studies as they argue that effective user documentation is not just a matter of substituting more user-friendly words for technical ones and prettying up a page design. Rather, in these studies, creating procedural discourse was a process of transforming declarative knowledge into procedural knowledge. With a broader understanding of the complex factors that guide discourse decisions, technical communicators might argue for greater access to information about audience, organizational image and conditions, subject matter, and conventions, thus improving the quality of the procedural discourse for technology that affects many users.

Understanding that technical communicators create meaning may also help to elevate the status of technical communicators' organizational roles and of the discourse that they create. Such an improvement in status can prevent some of the problems that technical communicators express, such as that they sometimes must spend a disproportionate amount of time and energy persuading coworkers of the worth of technical communication (Wilson & Ford, 2003). A greater appreciation of the importance of technical communicators' work can encourage organizations to provide technical communicators with resources to support the development of their understanding of users, organizational conditions, technology, and conventions so that they can create effective procedural knowledge for end users.

Teachers and mentors of novice technical communicators should also assist their students in developing the skills required to create a deep understanding of these four factors. With a rich understanding of users, organizational conditions, software, and genre conventions, technical communicators have the potential to create user discourse that contributes significantly to users' procedural knowledge.

Implications for Future Research

Although these studies provide insights into multiple factors that influence the creation of procedural discourse, future studies might address the following questions:

- What are the relative levels of influence in other settings of the four factors that guided the technical communicators' discourse decisions in these studies? How do technical communicators within other contexts shape their visions of these areas? How do technical communicators in other contexts create procedural knowledge?
- If, as Slack et al. (1993) argued, technical communicators who create meaning have additional ethical responsibilities for the texts, how do they handle these responsibilities? Do they "analyze critically the ethical implications of the meanings they contribute to" (p. 32) as they make decisions? If so, how?
- As we move into global contexts, how do technical communicators form and use mental models of users and contexts in other cultures through direct and indirect means (Warren, 1994, p. 176)? How do they employ their understandings of subject matter, context, and form in cross-cultural settings?
- What can these findings contribute to general theories of the social and epistemic nature of many kinds of discourse?

Conclusion

Because poorly designed procedural discourse can frustrate users and alienate them from technology that could improve the quality of their lives, understanding how technical communicators create procedural meaning holds significant implications. Organizations that provide technical communicators with sufficient resources to allow them to create well-designed procedural discourse can empower users so that they can feel more in control of technology. "If we do not work to articulate rich techniques for invention in the education of technical communicators," as Regli (1999) argued, "we inadvertently reinforce the myth of the technical communicator as born scribe—a fortuitously gifted communicator who by instinct knows how to 'clean up' the products of the real 'inventors' of technical information" (pp. 31-32). The long-reaching impact of technology on contemporary lives requires respect for the complex process of creating procedural discourse. A rich understanding of how technical communicators function as authors in

making decisions about discourse reveals that they are not just "translating" or "transmitting" information. Insights from these studies continue the process of improving the quality of procedural discourse and knowledge.

Procedural knowledge is located within practice as well as within discourse, making it challenging to create. The factors contributing to its creation appear to be similar to those that contribute to the creation of other types of discourse. But the creation of procedural discourse is further complicated by the contexts in which it is created and interpreted and by the type of knowledge with which it deals. Because of its complexity, the creation of procedural discourse and knowledge is a rich field, worthy of respect and ripe for continued exploration.

Acknowledgments

Thanks to Ed Nagelhout, the participants in an Association of Teachers of Technical Writing Research Network Forum, and the anonymous *JBTC* reviewers for their comments on drafts of this article.

Declaration of Conflicting Interests

The author declared no conflicts of interest with respect to the authorship and/or publication of this article.

Funding

This research was supported in part by grants from the Purdue School of Engineering and Technology, Indianapolis.

References

Applen, J. D. (2002). Technical communication, knowledge management, and XML. *Technical Communication*, *49*, 301-313.

Atwill, J. M. (1998). *Discourse reclaimed: Aristotle and the liberal arts tradition.* Ithaca, NY: Cornell University Press.

Backman, M. (1987). Introduction: Richard McKeon and the renaissance of discourse. In M. Backman (Ed.), *Discourse: Essays in invention and discovery* (pp. vii-xxxii). Woodbridge, CT: Ox Bow Press.

Barnum, C. M. (2002). *Usability Testing and Research.* New York: Longman.

Charney, D. H., Reder, L. M., & Wells, G. W. (1988). Studies of elaboration in procedural texts. In S. Doheny-Farina (Ed.), *Effective documentation: What we have learned from research* (pp. 47-72). Cambridge, MA: MIT Press.

Farkas, D. K. (1999). The logical and rhetorical construction of procedural discourse. *Technical Communication*, *46*, 42-66.

Farrell, T. B. (1976). Knowledge, consensus, and rhetorical theory. *Quarterly Journal of Speech*, *62*, 1-14.

Hanks, W. F. (1991). Foreword. In J. Lave & E. Wenger (Eds.), *Situated learning: Legitimate peripheral participation* (pp. 13-24). Cambridge, UK: Cambridge University Press.

Hargis, G. (1998). *Developing quality technical information: A handbook for writers and editors.* Upper Saddle River, NJ: Prentice Hall.

Henry, J. (1998). Documenting contributory expertise: The value added by technical communicators in collaborative writing situations. *Technical Communication*, *45*, 207-220.

Henry, J. (2000). *Writing workplace cultures: An archaeology of professional writing.* Carbondale: Southern Illinois University Press.

Herrington, A. J. (1985). Writing in academic settings: A study of the contexts for writing in two college chemical engineering courses. *Research in the Teaching of English*, *19*, 331-359.

Hovde, M. R. (2000). Tactics for building images of audience in organizational contexts: An ethnographic study of technical communicators. *Journal of Business and Technical Communication*, *14*, 395-444.

Hovde, M. R. (2001). Research tactics for constructing perceptions of subject matter in organizational contexts: An ethnographic study of technical communicators. *Technical Communication Quarterly*, *10*, 175-211.

Hovde, M. R. (2002). Negotiating organizational constraints: Options for technical communicators. *Technostyle*, *18*, 61-94.

Hovde, M. R., & Hovde, D. M. (2002). Elements of the expertise technology trainers need to instruct academic library employees. *Portal: Libraries and the Academy*, *2*, 601-625.

Hughes, M. (2002). Moving from information transfer to knowledge creation: A new value proposition for technical communicators. *Technical Communication*, *49*, 275-285.

Johnson, R. R. (1997). Audience involved: Toward a participatory model of writing. *Computers and Composition*, *14*, 361-376.

Johnson, R. R. (1998). *User-centered technology: A rhetorical theory for computers and other mundane texts.* Albany: State University of New York Press.

Johnson-Eilola, J. (1997). *Nostalgic angels: Rearticulating hypertext writing.* Norwood, NJ: Ablex.

Karreman, J. (2004). *Use and effect of declarative information in user instructions.* Amsterdam: Rodopi.

Kinneavy, J. (1971). *Theory of discourse: The aims of discourse.* Englewood Cliffs, NJ: Prentice Hall.

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation.* Cambridge, UK: Cambridge University Press.

Mardsjo, K. (1994). Man–Text–Technology: Technical manuals as a means of communication. In M. Steehouder, C. Jansen, P. van der Poort, & R. Verheijen (Eds.), *Quality of technical documentation* (pp. 185-200). Atlanta, GA: Rodopi.

Mathes, J. C., & Stevenson, D. W. (1991). *Designing technical reports: Writing for audiences in organizations* (2nd ed.). Needham Heights, MA: Bobbs-Merrill.

Paradis, J. (1991). Text and action: The operator's manual in context and in court. In C. Bazerman & J. Paradis (Eds.), *Textual dynamics of the professions: Historical and contemporary studies of writing in professional communities* (pp. 256-278). Madison: University of Wisconsin Press.

Paré, A., & Smart, G. (1994). Observing genres in action: Towards a research methodology. In A. Freedman & P. Medway (Eds.), *Genre and the new discourse* (pp. 146-154). Washington, DC: Taylor and Francis.

Porter, J. E. (1992). *Audience and discourse: An archeological composition of the discourse community.* Englewood Cliffs, NJ: Prentice Hall.

Price, J., & Korman, H. (1993). *How to communicate technical information: A handbook of software and hardware documentation.* Redwood City, CA: Benjamin/Cummings.

Raven, M. E. (1992). *Analyzing and adapting to multiple audiences: A study of two writers in the computer industry.* Unpublished doctoral dissertation, Rensselaer Polytechnic Institute, Troy, NY.

Regli, S. H. (1999). Whose ideas? The technical writer's expertise in "*inventio*.". *Journal of Technical Writing and Communication*, *29*, 31-40.

Salvo, M., Zoetewey, M. W., & Agena, K. (2007). A case of exhaustive documentation: Re-centering system-oriented organizations around user need. *Technical Communication*, *54*, 46-57.

Selzer, J. (1983). The composing process of an engineer. *College Composition and Communication*, *34*, 178-187.

Simon, H. W. (1990). The discourse of inquiry as an intellectual movement. In H. W. Simon (Ed.), *The rhetorical turn: Invention and persuasion in the conduct of inquiry* (pp. 1-34). University of Chicago Press.

Slack, J. D., Miller, D. J., & Doak, J. (1993). The technical communicator as author: Meaning, power, authority. *Journal of Business and Technical Communication*, *7*, 12-36.

Smart, G. (1999a). Reinventing expertise: Experienced writers in the workplace encounter a new genre. In P. Dias & A. Paré (Eds.), *Transitions: Writing in academic and workplace settings* (pp. 223-252). Mahwah, NJ: Erlbaum.

Smart, G. (1999b). Storytelling in a central bank: The role of narrative in the creation and use of specialized economic knowledge. *Journal of Business and Technical Communication*, *13*, 249-273.

Smart, K. L., Madrigal, J. L., & Seawright, K. K. (1996). The effect of documentation on customer perception of product quality. *IEEE Transactions on Professional Communication*, *39*, 157-162.

Smith, H. T., & Shirk, H. N. (1996). The perils of defective documentation: Preparing business and technical communicators to avoid product liability. *Journal of Business and Technical Communication*, *10*, 187-202.

Spivey, N. N. (1997). *The constructivist metaphor: Reading, writing, and the making of meaning.* San Diego, CA: Academic Press.

Subbiah, M. (1997). Social construction theory and technical communication. In K. Staples & C. Ornatowski (Eds.), *Foundations for teaching technical communication: Theory, practice, and program design* (pp. 53-65). Greenwich, CT: Ablex.

Suchman, L. (1987). *Plans and situated actions: The problem of human-machine communication.* New York: Cambridge University Press.

Warren, T. L. (1994). Issues in internationalization of documentation: Quality control. In M. Steehouder, C. Jansen, P. van der Poort, & R. Verheijen (Eds.), *Quality of technical documentation* (pp. 170-184). Atlanta, GA: Rodopi.

Wells, S. (1986). Jürgen Habermas, communicative competence, and the teaching of technical discourse. In C. Nelson (Ed.), *Theory in the classroom* (pp. 245-269). Urbana: University of Illinois Press.

Wilson, G., & Ford, J. D. (2003). The big chill: Seven technical communicators talk ten years after their master's program. *Technical Communication*, *50*, 145-159.

Winsor, D. A. (1989). An engineer's writing and the corporate construction of knowledge. *Written Communication*, *6*, 270-285.

Wright, P. (1994). Quality or usability? Quality writing provokes quality reading. In M. Steehouder, C. Jansen, P. van der Poort, & R. Verheijen (Eds.), *Quality of technical documentation* (pp. 7-37). Atlanta, GA: Rodopi.

Bio

Marjorie Rush Hovde is an associate professor of technical communication and English at Indiana University Purdue University Indianapolis. Her research focuses on empirical studies of technical communicators as they create meaning in discourse within their workplaces.

Fonte: Journal of Business and Technical Communication, V. 24, p. 164-205, 2010. [Base de Dados]. Disponível em: <http://online.sagepub.com>. Acesso em: 4 maio 2010.