Chapter 5: Procedural Knowledge and Discourse in Technical Communication: Easy as 1, 2, 3?

Marjorie Rush Hovde
Indiana University-Purdue University Indianapolis

Abstract: Within technical communication, understanding the complexities of procedural knowledge and discourse is crucial to creating effective user documentation in many forms. In addition to providing insights into procedural knowledge, this chapter explores differences between descriptive technical discourse and procedural technical discourse that helps people gain procedural knowledge. The chapter also explores several implications of these differences for creating effective procedural discourse, including the importance of usability testing of instructions, followed by a discussion addressing several myths about the creation of and importance of procedural discourse. The chapter closes with implications for future research into procedural knowledge and discourse.

Keywords: procedural discourse, procedural knowledge, instructions, documentation usability

I recently encountered a product whose label proclaimed that setup and use were as "Easy as 1, 2, 3." This phrase appeared to function as a marketing tool to persuade users to purchase a product that would be easy to use. However, in many technical contexts, processes are frequently more complex than "1, 2, 3" and can often frustrate and alienate users who do not know how to complete them. Rich understandings of the complexity of procedural knowledge and its discourse can help technical communicators navigate the challenges that arise when they try to teach users how to use technology or other systems effectively.

Whether technical communicators create stand-alone manuals, online help, training experiences, instructional videos, or other forms of procedural discourse intended for users, they benefit from understanding the complexities of procedural knowledge and its relevant discourse. Because "documentation is a learning medium that can transform the user experience, providing useful and practical information presented in a context-sensitive format" (Hogan, 2013, p. 156), paying close attention to communication can help to develop users' procedural knowledge. In addition, technical communicators are helped by understanding effective processes for creating procedural discourse, processes undergirded by foundational assumptions about the relationships between users and systems (Johnson, 1998).

DOI: https://doi.org/10.37514/TPC-B.2022.1381.2.05

However, many technical communicators (and their colleagues) may not be aware of the differences between descriptive knowledge and procedural knowledge, and thus they produce discourse that does not help users understand and follow relevant processes (Hovde, 2010). Furthermore, many forms of procedural discourse may be too simple for helping people to function within complex and interlocking systems.

This chapter explores the following:

- perspectives on procedural knowledge and procedural discourse,
- the complexities of thinking about technical communication for non-routine processes in complex contexts,
- practical implications for technical communicators who wish to improve processes for producing effective procedural discourse,
- the role of usability testing of procedural discourse, and
- several myths about the role of procedural discourse.

In addition to exploring my observations and experiences over many years, I draw on insights from a variety of scholars who provide rich understandings for practitioners as well as instructors and scholars of technical communication who wish to understand the richness and complexity of procedural knowledge, understandings that are foundational to creating procedural discourse.

Perspectives on Procedural Knowledge, Procedural Discourse, and Descriptive Discourse

Before exploring the implications of procedural discourse for technical communication, definitions of procedural knowledge, procedural discourse, and descriptive discourse may prove helpful.

Procedural Knowledge

Procedural knowledge exists in action. It typically begins in a situation where a current state is not desirable, includes actions that move toward a goal, and ideally ends when the goal state is achieved (Farkas, 1999). This knowledge "is not just cognitive, but often tactile and visual as well, relying on cues from context on when to act and what to do" (Durack, 1997). Procedural knowledge is a larger category than procedural discourse, but procedural discourse is essential, especially when people are learning to use a complex system.

Procedural knowledge combines "how-to" skills with conceptual knowledge of a system, sometimes called "knowing that." A system is a structure in which users need to work to achieve their goals. It may be a computer system, an organization, a device, an electronic game (deWinter, 2014), or a set of policies. Routine processes are usually easy to learn and remember. For instance, in withdrawing cash from an automatic teller, users insert a card, enter a PIN code, and select an option from the menu on a screen. The processes become slightly more complicated when users wish to check a balance, make a deposit, or transfer money from one account to another. However, even those processes are easy to learn and remember (or figure out if the screen interface is well designed).

Procedural knowledge can be clear cut and uncomplicated when users follow a routine of unvarying steps in a simple system, but it becomes more challenging when users need to follow multiple possible pathways (Albers, 2004; Roochnik, 1996; Swarts, 2014, 2015) to achieve their inter-connected web of goals (Albers, 2004) in complex systems. When routine actions are not possible, owing to contextual factors or combinations of complex systems, users need to think of alternative actions (Farkas, 1999). Goals may shift and emerge as users are trying to create their procedural knowledge. A recent example of insufficiently developed procedural knowledge is the cases of the two Boeing 737 MAX airplanes that crashed because the procedural knowledge of the pilots was inadequate for overcoming problems with new software, primarily because the retraining of experienced pilots proved inadequate for this complex situation (Associated Press, 2019).

People's goals in using a system often spring from their unique contexts, complicated by the fact that users may think in terms that may not be the same as the system's terminology (Mirel, 1993). User goals relate to their contexts and work patterns, involving the "user's mental process" (Albers, 2004, p. 79) more than the possible functions of a system. In addition, users face cognitive, environment, and technology constraints as they work with a system, issues that system designers may not have considered.

People develop procedural knowledge through a variety of approaches. Some users learn through trial-and-error explorations of a system (Mirel, 1993). Others learn through direct instruction combined with practice. Developing procedural knowledge frequently involves multiple senses, according to neuroscientists who argue, "learning and cognition are multi-sensory experiences" (Remley, 2015, p. vii), indicating that multiple parts of the brain are involved. Users may have a variety of strategies for learning—strategies that involve the mind, but also other sensory-motor experiences; procedural knowledge is gained through cognitive, social, and physical means.

Developing procedural knowledge may involve one-on-one interactions with experts or it may involve group training. In the medieval guild system and into the 19th century, apprentices and learners developed procedural knowledge through oral instruction and by imitating what their masters or parents did (Durack, 1998). Early 20th century military training also involved demonstrations, explanations, repetition, and hands-on practice (Remley, 2015, p. 71), so procedural knowledge was transmitted both orally and via practice. Frequently, oral-dominant cultures transmit knowledge differently than literate cultures do (Durack, 1997). (Procedural knowledge also has been considered in some circles to be of a low status and an inferior form of knowledge. This perception continues today—at one university with which I am familiar, students are encouraged to take "knowledge" courses rather than "how-to" courses. For instance, a course in the history of art is acceptable as an elective, but a course in creating art is not acceptable.)

Because time and cost may make synchronous one-on-one or group training prohibitive for helping people develop procedural knowledge, technical discourse can help. However, written documentation and other asynchronous forms of discourse for users also present limitations because their "inherent linearity and rigidity . . . coupled with the necessary reduction of complex situations to sequential units of simple action" may cause users to misunderstand the effective and safe use of a system or tool (Paradis, 1991). Because of the asynchronous nature of much procedural discourse, creators and users may be operating with differing assumptions (van Loggem, 2013). However, despite the limitations of asynchronous procedural discourse, developing procedural knowledge with the aid of discourse is usually more effective than having users learn processes solely through trial and error.

In addition to a person knowing "how" to work within a system, that person's conceptual or descriptive knowledge of a system plays an important role in developing procedural knowledge (Hovde & Renguette, 2017; Swarts, 2018), especially needed when troubleshooting, completing non-routine tasks, or learning new processes. For instance, in the days before graphical user interfaces, I learned of one user who rebooted his computer each time he wanted to escape something. He did not know that a key on his keyboard would allow him to go back to a previous screen, basic conceptual knowledge about a toggle option that would have saved him a great deal of time. Conceptual knowledge, however, is not sufficient for achieving procedural knowledge. For instance, learning music theory can be helpful when learning to play a new instrument, but instruction and practice are needed to produce music. Moving through non-routine processes will require users who possess enough conceptual knowledge to know what to do when conditions shift (Farkas, 1999); conceptual knowledge provides an important foundation when tackling non-routine and/or complex processes.

Procedural knowledge within a person changes over time. When beginning to learn a process, users may need to learn through explicit steps. However, over time and with practice, procedural knowledge becomes internalized and tacit, seeming like second nature to the actor. For instance, a novice may perceive saving a file as several discrete steps, whereas a more experienced user will conceive of the process as a step or two. Although beginners may start with simple, clear-cut procedural knowledge, they often move to addressing problems that are "murky, unpredictable, and uncertain" (Swarts, 2018, p. 38). Complex non-routine processes are more challenging to learn and remember (Albers, 2004; Swarts, 2018), and thus conceptual knowledge plays an important role in learning and memory. For instance, a person moving to a new city may need a map which provides descriptive information that allows for navigation. However, once that person has become familiar with the streets, that person can often figure out the best route, using conceptual knowledge gained through experience and observation.

Overall, procedural knowledge involves knowing how to complete tasks in order to achieve goals. It encompasses both knowing "how" and knowing "that." In addition, procedural knowledge involves possessing enough conceptual knowledge to improvise when non-routine situations arise. Procedural knowledge exists in the doing, so it is difficult to capture solely in discourse. People gain procedural knowledge through instruction and practice, which may include trial and error as well as multiple sensory experiences. Technical communication in many forms at its best functions to "help accommodate technologies and texts to our situated use" (Swarts, 2018, p. 3). Because procedural knowledge exists in the doing and within users' physical bodies and memories, capturing and describing procedural knowledge can be challenging. Although procedural discourse is not the same as procedural knowledge, discourse plays an important role in developing procedural knowledge within individuals and within communities, as discussed in the next section.

Procedural Discourse

Procedural discourse is intended to help people accomplish goals (Farkas, 1999) in relation to a system and to develop their procedural knowledge. The system may be technical, related to an organization, or related to a larger network of resources and actions. For instance,

- Online help can assist users in employing software for their purposes.
- An employee manual can help users figure out how to function within their organization.
- An agricultural manual can help work within "a network of constructed waterways, the knowledge of when and how to irrigate fields, and the entire set of human activities that comprise this method for farming" (Durack, 1997, p. 258).
- Manuals or in-game instructions can assist people in playing electronic games (deWinter, 2014).

Procedural discourse plays an important role in creating larger and more complex procedural knowledge.

Over time, humans have devised a number of forms of procedural discourse which can involve more than words, encompassing a variety of symbol systems including the visual (Remley, 2015; Tenbrink & Maas, 2015). Procedural discourse may take forms such as paper manuals, training sessions, how-to videos, online help, or informal conversations among users, face to face or online. Online forums have the advantage of crowd sourcing, drawing on the resources and experiences of many users to address non-routine uses of a system. This form of user support becomes a conversation or dialog that can adapt to unique needs. These dialogic approaches not only answer questions, but also help users develop abilities to solve future problems (Swarts, 2018, p. 72). Having access to a variety of types of procedural discourse can allow adaptation to unique circumstances and a range of learning styles. Because many technical communicators also design websites that provide content other than online help, understanding procedural discourse assists in creating user interfaces that allow users to move easily through the tasks they need to accomplish on websites.

Whatever forms it takes, procedural discourse is necessarily a simplified version of procedural knowledge and is designed to assist users in learning (Paradis, 1991). Procedural discourse can allow for organizing knowledge and provide a means of sharing knowledge with others distant in time and space. Traditionally, software documentation has been aimed at "the normalization of user behavior . . . to teach the users what the software is capable of doing, how it can be done, and what are the best practices" (Swarts, 2018, p. 100). As they learn, users gradually develop procedural knowledge.

Because the brain changes as it learns new tasks, multimodal instructional materials—"print-linguistic, visual, audio, gestural, and spatial" (Remley, 2015, p. 24)—are crucial and help users learn and remember material because of the reinforcement from multiple senses. Overall, the effectiveness of multimedia may depend on users' learning styles and prior experiences (Remley, 2015, p. 37). Furthermore, if a user has a biological limitation, gaining procedural knowledge may be adapted to take that limitation into account (Albers, 2004; Remley, 2015). Imitation and practice are key to learning new processes, so procedural discourse alone is likely to be inadequate in developing procedural knowledge; nevertheless, the discourse can play a significant role for users.

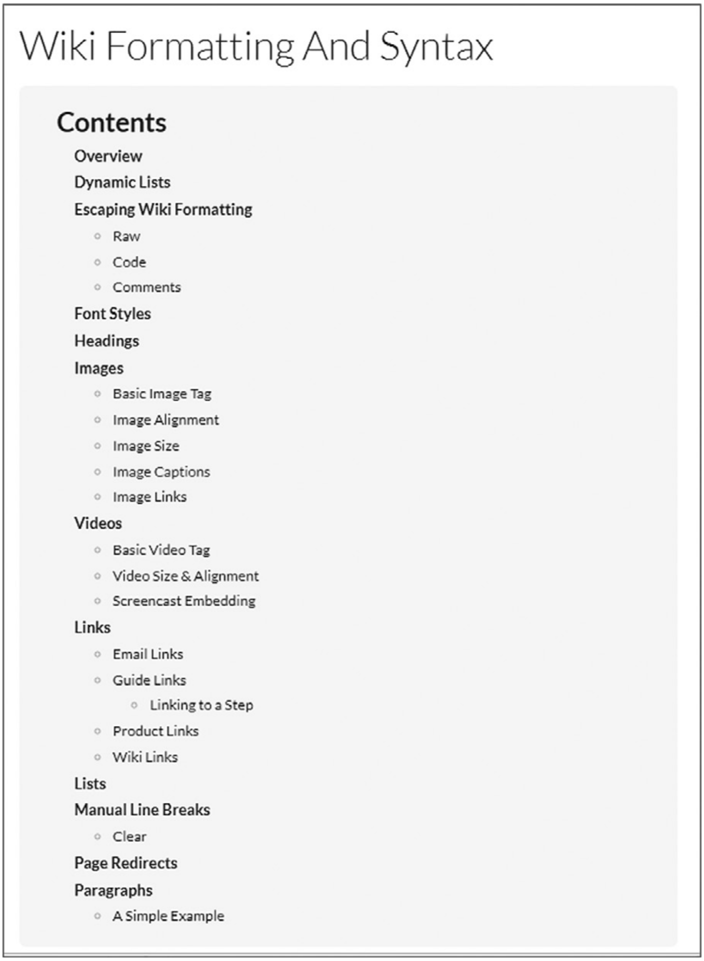
Effective user documentation has significant social effects because this discourse can "interpret for the lay public the meanings, application, and procedures by which expert products . . . are integrated into the behavioral flow of society itself" (Paradis, 1991, p. 256), thus lowering barriers to access to sophisticated technological systems.

Because discourse can help users attain procedural knowledge, various approaches, especially in written guides, have emerged, some more helpful than others. In looking at approaches to procedural discourse, especially user documentation, one usually encounters several varieties: system-oriented discourse, user-friendly discourse, mixed system and user-task discourse, and user task-oriented discourse, as discussed below.

System-Oriented/Descriptive Discourse

System-oriented/descriptive discourse focuses on describing the features of a system and is most helpful at developing conceptual knowledge but is severely lacking in its ability to develop users' procedural knowledge. The most common format is technical specifications that describe the architecture of a system or product. Descriptions do not include "how-to" information, so users have to extrapolate how to use a system (Hovde, 2010). For example, in the early 1980s, when I was learning to use the word processing program Wylbur on an IBM mainframe, the system-oriented documentation consisted of a ten-foot shelf of papers in no apparent order on one wall of the computer center. At times, users walked over to it and looked up information, but most of us learned to use Wylbur either by asking people near us how to complete tasks and/or through trial and error. The system-oriented guides were useful only to a few people who had appropriate background and who could navigate the materials.

Figure 5.1 illustrates descriptive discourse that focuses on a system. Simply reviewing the table of contents will not let users know what tasks or goals they might accomplish using this publication. Most of the items in the list are nouns or noun phrases, indicating that the documentation describes the system features rather than how to use the system. Such wording is not as helpful to users as verb phrases that indicate user actions (Farkas, 1999, p. 46).



Wiki Formatting And Syntax

Contents

- Overview
- Dynamic Lists
- Escaping Wiki Formatting
  - Raw
  - Code
  - Comments
- Font Styles
- Headings
- Images
  - Basic Image Tag
  - Image Alignment
  - Image Size
  - Image Captions
  - Image Links
- Videos
  - Basic Video Tag
  - Video Size & Alignment
  - Screencast Embedding
- Links
  - Email Links
  - Guide Links
    - Linking to a Step
  - Product Links
  - Wiki Links
- Lists
- Manual Line Breaks
  - Clear
- Page Redirects
- Paragraphs
  - A Simple Example

Figure 5.1. Descriptive documentation on the iFixit wiki help page. (Source: https://www.ifixit.com/Help/Wiki_Formatting_And_Syntax by permission of Creative Commons BY-NC-SA 3.0)

One problem with descriptive system-oriented technical discourse is that it may easily become "exhaustive," impeding usability by including information (Remley, 2015) that users may not need. For example, the documentation featured in Michael Salvo et al.'s (2007) study included complete details of a system, but it included much more information than users needed. Figure 5.2 shows an exploded diagram that describes the parts of the system but does not let users know about relevant processes for installing or using the system.

Exhaustive, system-oriented documentation typically ignores users' needs and perceptions, focusing attention on describing a structure (Johnson, 1998). Unsurprisingly, if users see only a static conceptual description, they typically find it difficult to use that system. Some users may try to learn to use a system through trial and error, but most ordinary users do not have the conceptual background, time, or patience to learn through that means.

Another problem with system-oriented documentation is that conflicts between clarity for the reader versus completeness of information about the system may arise because "with the information both hard to find and hard to process, the communication between the interface and the user has broken down and, for all practical purposes, the information doesn't exist" (Albers, 2004, p. 110). Additionally, it is almost impossible to provide complete information about a system; technical communicators need to decide what to include and what to exclude. Ideally, user documentation provides enough information to help users meet their goals but not so much that users become overwhelmed and cannot determine what information is relevant.

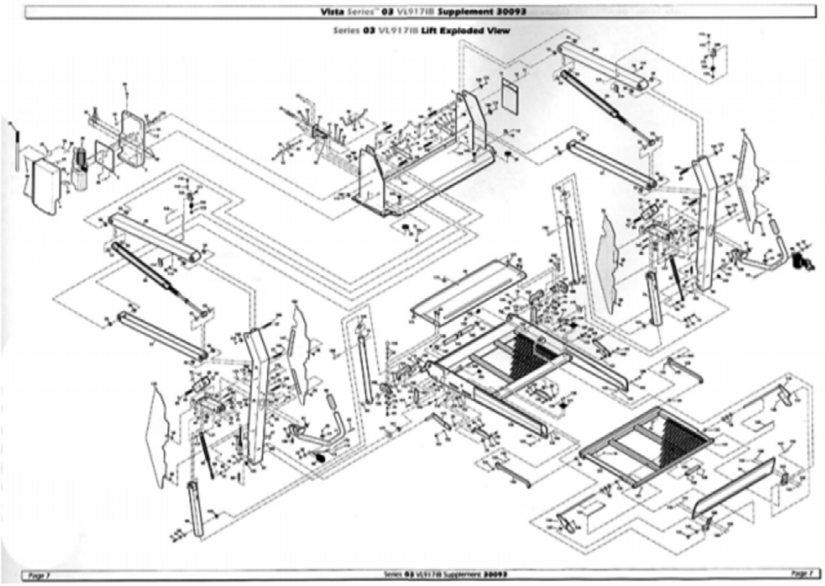

Vista Series 03 VL917IB Supplement 30093
Series 03 VL917IB Lift Exploded View
Page 7
Series 03 VL917IB Supplement 30093
Page 7

Figure 5.2. Exploded diagram (Salvo et al., 2007, p. 51) describing a system in exhaustive detail. (Used by permission of the Society for Technical Communication.)

User-Friendly Documentation

To assist non-specialist users, some creators of technical communication attempt to make documentation user-friendly by using simple, informal language and attractive visual formatting, but "user-friendly" is not the same as user-task-oriented (Johnson, 1998). For example, when Google introduced Chrome in 2008, they released a comic book style explanation of the new browser, using drawings of people with word bubbles and casual, simple language explaining why this browser was unique, as excerpted in Figure 5.3. While the visuals and the language level make the information accessible to an audience with limited technical background, the publication was not focused on how to use the new browser; instead, its content dealt with the logic behind the design and important features of the system.

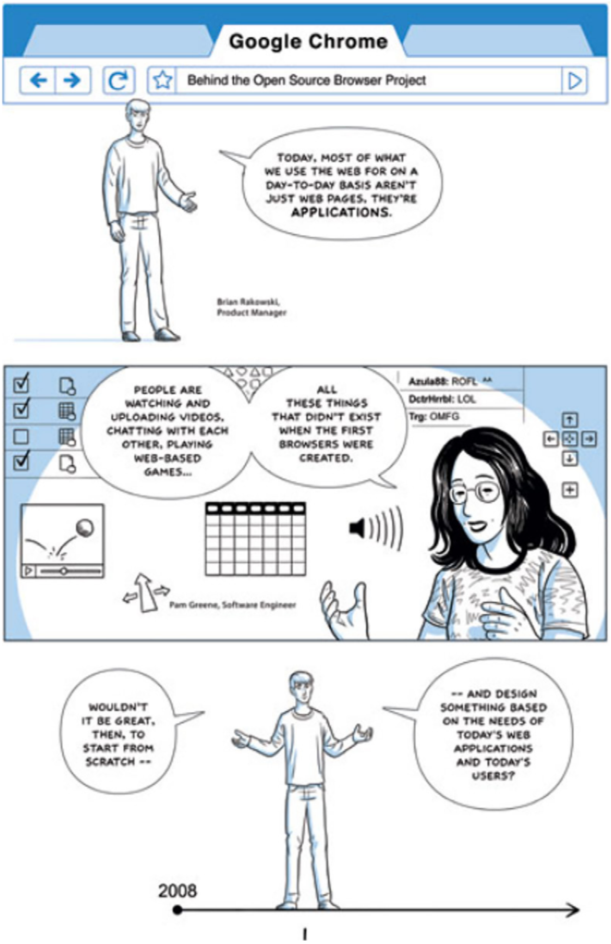
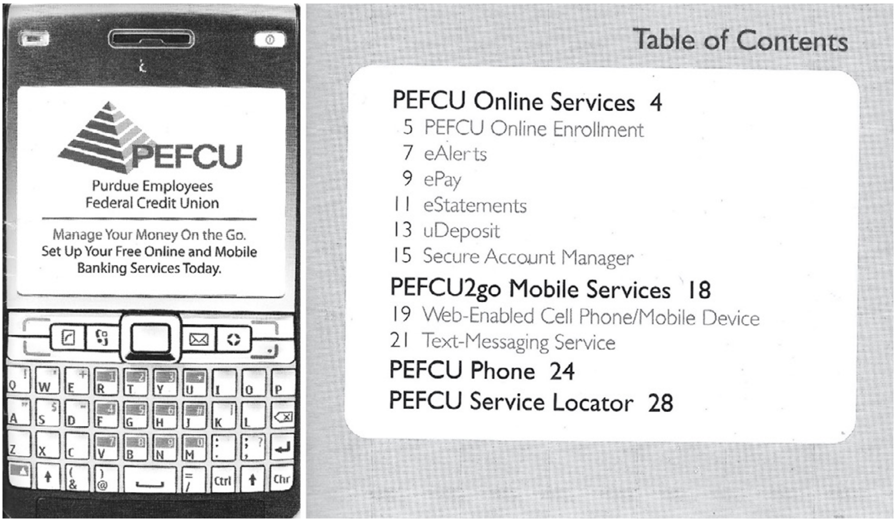

Google Chrome
Behind the Open Source Browser Project
TODAY, MOST OF WHAT WE USE THE WEB FOR ON A DAY-TO-DAY BASIS AREN'T JUST WEB PAGES, THEY'RE APPLICATIONS.
Brian Rakowski, Product Manager
PEOPLE ARE WATCHING AND UPLOADING VIDEOS, CHATTING WITH EACH OTHER, PLAYING WEB-BASED GAMES...
ALL THESE THINGS THAT DIDN'T EXIST WHEN THE FIRST BROWSERS WERE CREATED.
Azula88: ROFL ^^
DctrHrrbl: LOL
Trg: OMFG
Pam Greene, Software Engineer
WOULDN'T IT BE GREAT, THEN, TO START FROM SCRATCH --
-- AND DESIGN SOMETHING BASED ON THE NEEDS OF TODAY'S WEB APPLICATIONS AND TODAY'S USERS?
2008

Figure 5.3. A user-friendly approach is not the same as a user-task-oriented approach, as illustrated by this excerpt from the Google Chrome comic book (McCloud, 2008; Creative Commons Attribution-Noncommercial-No Derivative Works 2.5 License).

Another example comes from a credit union during the earlier days of mobile phones. The user guide consisted of a small booklet that was the size and shape of a mobile phone, as seen in Figure 5.4, and it provided information about the credit union's app for accessing online services. However, while the booklet was user-friendly, it was not user-task-oriented, as seen in its table of contents in Figure 5.4, and thus not overtly helpful for using the bank's online services.

A user-friendly approach often presents a system orientation rather than a user-task orientation (Johnson, 1998). While ordinary language may enhance readability, and personable discourse and visuals are appealing, a publication that is system-oriented but user-friendly does not provide users with a sense of the actions they might complete in relation to the system. In addition, a user-friendly publication may assume that users will read the text in a linear fashion, but learning through doing is often not linear (Johnson, 1998).

One form that user-friendly documentation often takes is a "Frequently Asked Questions" list. While these lists attempt to meet users where they are, such lists are often focused on the system rather than actual user needs (Albers, 2004). In addition, these lists are often poorly organized, forcing users to sift through many questions in order to get to the relevant answer, if they are able to do so at all. Simply providing information is not adequate if it is not shaped and presented in ways that are accessible to users (Albers, 2004). Some creators of documentation realize this limitation and blend system orientation and user-task orientation, an approach that has limitations of its own, as described in the next section.



PEFCU
Purdue Employees
Federal Credit Union

Manage Your Money On the Go.
Set Up Your Free Online and Mobile Banking Services Today.

Table of Contents

PEFCU Online Services 4
5 PEFCU Online Enrollment
7 eAlerts
9 ePay
11 eStatements
13 uDeposit
15 Secure Account Manager
PEFCU2go Mobile Services 18
19 Web-Enabled Cell Phone/Mobile Device
21 Text-Messaging Service
PEFCU Phone 24
PEFCU Service Locator 28

Figure 5.4. Although the cover of this guide employs a user-friendly format, the table of contents remains system oriented. (Purdue Employees Federal Credit Union, 2008. Photo taken by author.).

Blended System Orientation and User-Task Orientation

At times, user documentation mixes system orientation with user-task orientation, as illustrated in Figure 5.5, a table of contents from an older modem manual. The headings sometimes begin with gerunds indicating user tasks, but at other times, headings consist of nouns or noun phrases that occasionally include technical terminology that may be unfamiliar to novices. Furthermore, indentations suggest levels of hierarchy that indicate that the authors may not have been thinking in terms of the users' goals and tasks. Specifically, under the heading "Required Equipment," three user tasks are included at the end of the indented list where one would expect to see equipment items.

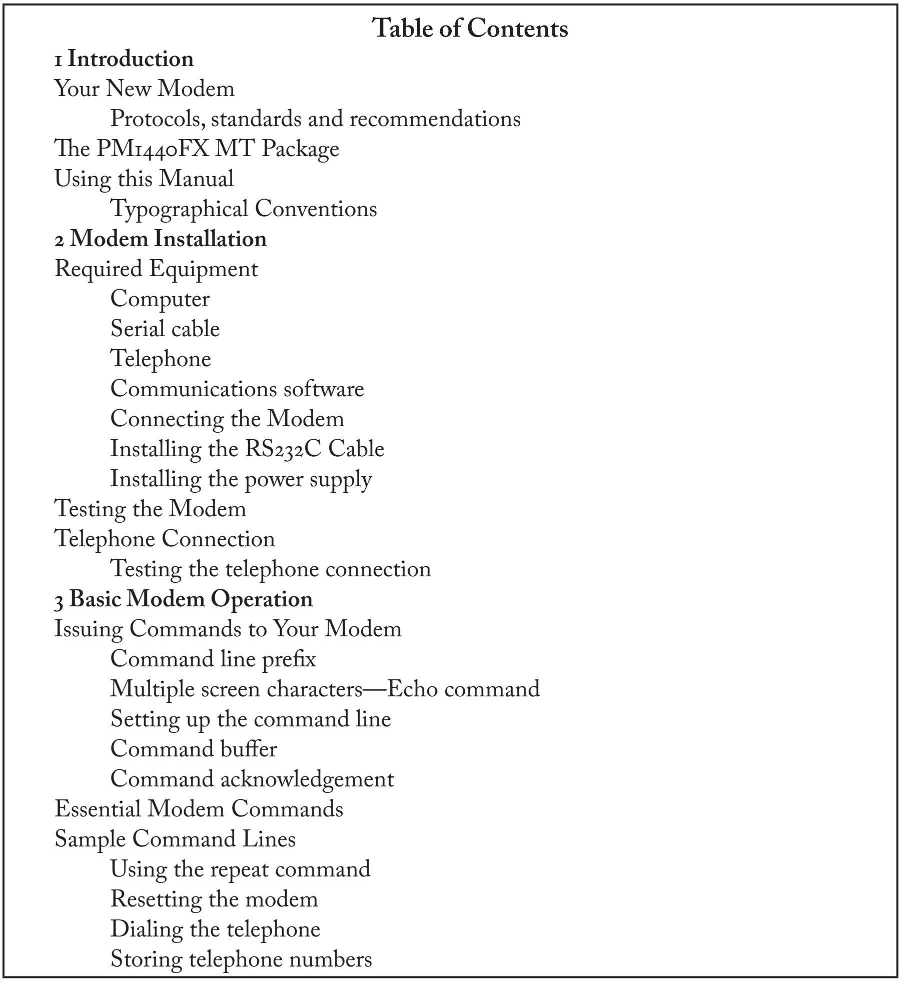


Table of Contents

1 Introduction

- Your New Modem
    - Protocols, standards and recommendations
- The PM1440FX MT Package
- Using this Manual
    - Typographical Conventions

2 Modem Installation

- Required Equipment
    - Computer
    - Serial cable
    - Telephone
    - Communications software
    - Connecting the Modem
    - Installing the RS232C Cable
    - Installing the power supply
- Testing the Modem
- Telephone Connection
    - Testing the telephone connection

3 Basic Modem Operation

- Issuing Commands to Your Modem
    - Command line prefix
    - Multiple screen characters—Echo command
    - Setting up the command line
    - Command buffer
    - Command acknowledgement
- Essential Modem Commands
- Sample Command Lines
    - Using the repeat command
    - Resetting the modem
    - Dialing the telephone
    - Storing telephone numbers

Figure 5.5. The opening of the table of contents of a modem manual with a mixture of procedural information and descriptive information. (From Practical Peripherals, 1993; recreated by author).

This modem documentation mixes system information with procedural information. The system-oriented headings ("Command line prefix," "Command buffer," etc.) do not give the users a sense of what they will learn to do in a given section. I speculate that the creators of this document were not aware of the differences between the two orientations within documentation and were not fully aware of the users' needs.

One line, "Using the repeat command," at first glance looks like a user task because it begins with a gerund. However, that line uses vocabulary focused on the system and does not indicate the goal (guided by context) that the user might have in using the repeat command. According to the manual, the command allows a previously entered command to be repeated, so a more user-task-oriented heading may be worded as "Repeating a previous command." Essentially, the creators probably did not consider that "The task is not in the software, and the user's purpose of interacting with the software is not to engage with it. . . . Instead, tasks live in the world" (Swarts, 2018, p. 29). System-oriented wording is minimally useful to end users.

Although this manual is older, I still frequently encounter this mixed approach in more recent manuals and user documentation, indicating that technical communication still has a long way to go to make sure that user documentation focuses on procedural discourse. (Fortunately, the back cover of this modem manual provides a number to call for tech support.) Overall, system information does not support user action (Albers, 2004). Discourse about a system is needed at times, but it does not meet the needs of people who want to develop procedural knowledge through user-task-oriented discourse, as discussed in the next section.

User-Task-Oriented Discourse

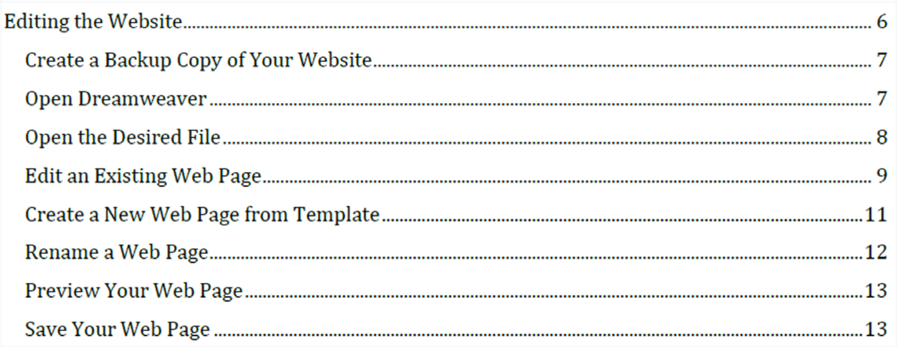
In contrast to the three approaches discussed above, user-task-oriented discourse focuses on how people use a system within their contexts. Technical communicators consider users' purposes and contexts as central to the decisions they make about user-task-oriented discourse. These instructions are typically focused on action adapted to users' situations (Johnson, 1998).

Figure 5.6 provides an excerpt from a user-task-oriented manual's table of contents. Each item begins with a verb, indicating what the user will learn to do in a given section. Although this list seems logical and useful, I rarely encounter this user-task-oriented approach in discourse that is intended to assist users in gaining procedural knowledge. Even if the wording looks user-task oriented, as mentioned in the previous section, the accompanying text may still be focused on the technology rather than the user (Durack, 1998).

Creating user-task-oriented documentation requires more skill than creating system-oriented documentation because in addition to knowing system information, technical communicators also need to know about users' knowledge levels, previous experience, and typical uses of the system relevant to their goals in context. In addition, technical communicators need to know the conventions of procedural discourse that users may expect to see or experience (Hovde, 2010; Paradis, 1991). User-task-oriented instructional discourse needs to "shift from the initial design and manufacturing orientation toward objects to a new orientation toward human thought and behavior" (Paradis, 1991, p. 176). Documentation that relies too much on the system's structure, even if the documentation is user-task oriented, may be too simple for experienced users who want to perform more complex tasks (Mirel, 1993). However, creators of user-task-oriented discourse also assume that users most likely do not need to know every feature of a system in order to use it.

User-task orientation is not inherently better than system orientation within technical communication; each has its function. A technical designer or developer needing technical specifications is unlikely to benefit from task-oriented user documentation, but users who need to know how to use a system are also unlikely to benefit from technical specifications that describe a system. Each approach has a communicative purpose, but when discourse is not designed appropriately for the communication situation, problems arise. User documentation that does not include procedural discourse and relevant conceptual knowledge (Paradis, 1991) may lead to wasted work and negative economic effects for a corporation when technical communicators describe a system in detail but neglect to consider the processes users follow when using the system (Salvo et al., 2007). In addition, poorly created instructional material may affect user safety, leading to liability issues for the organization that produced them (Hogan, 2013; Paradis, 1991; Remley, 2015). Technical communicators ideally aim to create accurate procedural discourse balanced with the level of detail needed by users. This goal is challenging because "there is not a fixed amount of information anyone needs, and different histories can change what a person needs" (Albers, 2004).

Table 5.1 summarizes several differences between user-task-oriented procedural discourse and descriptive discourse.



Editing the Website 6
Create a Backup Copy of Your Website 7
Open Dreamweaver 7
Open the Desired File 8
Edit an Existing Web Page 9
Create a New Web Page from Template 11
Rename a Web Page 12
Preview Your Web Page 13
Save Your Web Page 13

Figure 5.6. User-task orientation in a table of contents uses verbs and verb phrases to show users what actions they will learn in each section. (Author created example.)

Table 5.1. Key differences between descriptive discourse and procedural discourse

	Descriptive discourse	Procedural discourse
Purpose	▪ To describe all of the features of a system	▪ To teach users how to achieve goal states (Farkas, 1999) and interact with the system or technology ▪ To help users develop procedural knowledge
Scope	▪ Can be exhaustive (Salvo et al., 2007), describing the system in detail (Albers, 2004)	▪ Focuses mainly on tasks users need to complete, including conceptual/descriptive details only if they help in completing non-routine tasks
Intended audiences	▪ Technical designers or developers	▪ Users, installers, maintenance personnel
Ease of creation	▪ Relatively easy because a system exists and can be described (Salvo et al., 2007)	▪ More challenging because creators need to know about subject matter, audience, communication means, organizational constraints, and other situational variables (Farkas, 1999; Hovde, 2010; Johnson, 1998)
Markers of quality	▪ Accurate ▪ Thorough	▪ Effective ▪ Easy to use ▪ Memorable ▪ Efficient (Swarts, 2018) ▪ Useful for work beyond the system

Designing procedural discourse to teach users how to complete tasks within interlocked, networked systems is more complex than designing it for completing routine tasks (Albers, 2004; Swarts, 2015, 2018) within simple systems. Hence, technical communicators need to have a good understanding of procedural knowledge so that they can create effective procedural discourse for both routine and non-routine situations. However, they also need to be aware that user-task-oriented discourse is not as helpful "if it does not account for the vagaries of tasks in situ" (Swarts, 2018, p. 27) because user goals typically lie outside the system—in other words, using the system is not typically an end in itself.

Declarative or system knowledge may have a role to play in acquiring and practicing procedural knowledge, but conceptual knowledge works best if it is subordinate to a procedural structure and focus (Farkas, 1999; Karreman, 2004) in procedural discourse. Descriptive discourse such as technical specifications is usually not helpful to end users all by itself because of its focus on conceptual knowledge, which may be useful for troubleshooting or planning non-routine work. In addition, system designers or developers need descriptive discourse so that they can understand a system that they may have to modify or repair. However, not all users need all conceptual information (Salvo et al., 2007). Technical communicators need to understand the differences between conceptual knowledge/discourse and procedural knowledge/discourse and be able to design procedural discourse appropriately for its use. These differing approaches to procedural discourse spring from a history of assumptions and practices in creating user discourse, as discussed in the next section.

History of Assumptions About and Practices of Procedural Discourse Within Technical Communication

Jason Swarts (2018) provides a useful summary of changes in attitudes over the last several decades toward procedural discourse and the ways in which it was presented, and I will summarize that history here. Swarts notes that early 20th century understandings arose of the user manual as crucial to helping non-engineer audiences understand how to operate technology, especially in military contexts. Users of that era were typically not encouraged to vary from the instructions.

With the advent of computers and other advanced technology as early as the 1950s, documentation was frequently system oriented, and the focus remained on efficient use limited to how a system was designed. However, technologies made available to ordinary consumers also created a need for user guides to help them employ those technologies. For example, Figure 5.7 shows the table of contents from an old manual, probably from the 1940s, that includes about eight pages dealing with how to use an electric refrigerator and about 20 pages of menus and recipes, the latter topics no doubt intended for people who wanted to use this device in a well-run household and/or who were moving from an icebox to their first electric refrigerator. (For more on changing relationships between workplace and domestic technology and users, see Durack, 1997.)

This change in audience who had a range of "situated and experiential knowledge," frequently tacit (Swarts, 2018, p. 11), often led to the beginning of users' attempts to adapt technologies in ways that the designers did not intend. Because the goal was efficient use, the conventions of manuals focused on "simple and direct language, short sentences, active constructions, sequentially ordered steps, and a simple focus on one item/task at a time" (Swarts, 2018, p. 12).

In the 1980s, the concept of "Goals-Operations-Methods-Selection" (Mirel, 1993, p. 24; Swarts, 2018) emerged that equated user tasks with system tasks. Later, in the 1990s, understandings developed that user tasks and goals go well beyond system tasks to include "relationships among readers, text, tasks, interface designs, and exploratory types of problem-solving strategies" (Mirel, 1993, p. 25). Notably, "when user needs grew beyond the technology, the documentation served no clear knowledge creation function" (Swarts, 2018, p. 14). Thus, documentation was useful only for a limited set of tasks, but users pushed the boundaries of what software could do as those users became more knowledgeable about software's possibilities and experienced needs beyond those the documentation addressed.

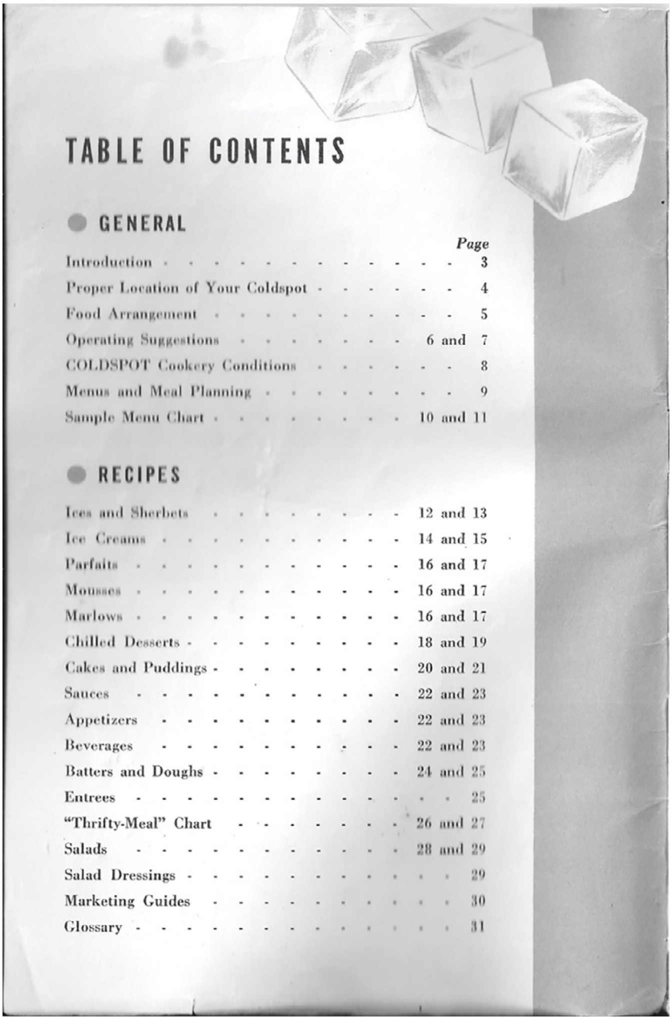


TABLE OF CONTENTS

● GENERAL

	Page
Introduction	3
Proper Location of Your Coldspot	4
Food Arrangement	5
Operating Suggestions	6 and 7
COLDSPOT Cookery Conditions	8
Menus and Meal Planning	9
Sample Menu Chart	10 and 11

● RECIPES

Ices and Sherbets	12 and 13
Ice Creams	14 and 15
Parfaits	16 and 17
Mousses	16 and 17
Marlows	16 and 17
Chilled Desserts	18 and 19
Cakes and Puddings	20 and 21
Sauces	22 and 23
Appetizers	22 and 23
Beverages	22 and 23
Batters and Doughs	24 and 25
Entrees	25
"Thrifty-Meal" Chart	26 and 27
Salads	28 and 29
Salad Dressings	29
Marketing Guides	30
Glossary	31

Figure 5.7. The table of contents from a vintage refrigerator manual includes about eight pages of technical details and about 20 of menus and recipes (Sears, Roebuck and Company, n.d. – photo taken by author).

Also in the 1980s, with the expansion of the availability of computers in workplaces and homes, manuals became more user-task oriented with conceptual information providing users a foundation for understanding the tasks (Swarts, 2018). During this time, controversy arose about whether comprehensive or minimalist manuals were better for learning (Karreman, 2004; Remley, 2015; Swarts, 2018). This minimalist documentation often encouraged users to explore and go beyond what typical documentation offered in using the technology (Swarts, 2018). Various studies indicated that a minimalist manual was preferable for learning by doing, but that users who had conceptual information from more comprehensive manuals did benefit, especially when presented with atypical situations of use (Remley, 2015). However, standard documentation still addressed the basics of operations; fewer means of supporting user learning and action beyond those standard processes were available (Swarts, 2018).

Newer forms of user documentation are currently emerging, but they have not become standardized and may not ever become standardized because of the complex network of user needs in the context of multiple software applications with which users are working. Technical communicators may need to be able to provide "interactive and dynamic help" (Swarts, 2018, p. 19) but also recognize that they are not the only people to create knowledge for users. Given the expanding nature of software and its use, procedural discourse may not be appropriately adapted to these new circumstances (van Loggem, 2013), so new approaches may need to emerge.

Ideally, the history of user documentation would show progress from system orientation to user-task orientation in discourse aimed at users, but I still find far too many examples of system-oriented documentation today. For instance, the documentation included in Figure 5.1 came from a relatively recent source, iFixIt.com, which provides a great deal of technical instructional material, so one would think that organization would understand the importance of user-task orientation. Alas, that is apparently not the case. Technical communicators need to find better means of applying recent advances in thinking about user discourse to actual practice and delivery. Understanding the history of and important concepts about procedural discourse holds many implications for the creation of effective procedural discourse today and in the future. However, additional insights about effective creation processes, as described in the next section, should be useful to technical communicators.

Creating Effective Procedural Discourse

In order to create effective, usable, and useful user documentation, creators of procedural discourse in its many forms need to understand not only the differences between system knowledge/discourse and procedural knowledge/discourse but also elements of an effective creation process. If they do not, they may produce unusable system-oriented exhaustive documentation that consumes a great deal of time and resources while being created, is not effective, and does not meet the needs of users. Furthermore, having engineers or marketing personnel (rather than technical communicators) create user documentation may lead to discourse that is not useful or is even unsafe for users (Paradis, 1991). Understanding procedural discourse and knowledge makes it possible to create user-task-oriented documentation (Salvo et al., 2007) that is more likely to help end users. Dávid Farkas (1999) has provided a foundational model of procedural discourse for technical communication on which others have built (Swarts, 2015, 2018), but many technical communication practitioners still struggle with creating "how-to" documentation that helps users complete tasks and processes. Creating effective procedural discourse takes conscious expertise and a context that encourages its creation.

Technical communicators need expertise in several areas in order to create effective procedural discourse. In addition to being aware of good processes, technical communicators also need understandings of the audience of users, authorial image/concerns, content selection, and genre conventions (Hovde, 2010), as well as considering the contextual variables, as explored below.

Effective Processes for Creating Procedural Discourse

User-task vs. system-oriented approaches can affect processes for creating technical communication. For instance, creating system-oriented discourse mainly involves getting to know the system well and not necessarily considering end users' needs and practices. If the discourse focuses on user-friendliness, then the creation process focuses on assuring readability. However, if the discourse focuses on user tasks and usability, the creation process begins with understanding users and their situations of use but also includes getting to know the system. Ideally, technical communicators participate early in the technical development process so that they can get to know the system as well as advocate for users when foundational decisions are made about the nature of the interface and user documentation (Johnson, 1998). Formative and summative usability testing of instructional material involving typical users can ensure that the procedural discourse meets its goal of developing procedural knowledge within users.

In addition to understanding users well, technical communicators may find that the role involves becoming "a facilitator or network maker, someone who is skilled at finding the right information and making the right connections and creating the right formats and protocols to meet the users' needs" (Swarts, 2018, p. 150) in relation to complex systems and networked technologies. In these contexts, technical communicators will organize content and make it easy for users to access as well as "managing the process of knowledge creation" (Swarts, 2018, p. 152). This function may go beyond the typical understandings of the roles and natures of technical communicators.

Another element in an effective process is focusing on the usability of the procedural discourse. Technical communicators need to consider more than the tasks that the system implies or is designed for. Instead, they need to consider contextual dynamics of users' work lives to determine if the instructions are useful to users (Durack, 1998; Mirel, 1993). While it is wise to design documentation that is readable, accurate, and has accessible information, technical communicators also need to analyze users' levels of job responsibilities and their desires to adapt systems to their work contexts (Mirel, 1993). Unfortunately, many technical communicators still lack access to rich information about users (Hovde, 2001), so developing these perspectives about users can prove challenging.

Although procedural discourse may be designed with usability in mind, well-designed usability testing of materials plays a crucial role in designing effective user support materials and in ascertaining their effectiveness. Technical communicators who understand the need for user-centered design test their discourse to see if it achieves its aims of teaching users "how-to" knowledge in an efficient and effective manner (Alexander, 2013). Because procedural knowledge is complex, this testing is crucial to help determine if user documentation has reached its goals. Much has been written about the usability of documentation and its testing (Barnum, 2011); a detailed discussion is beyond the scope of this chapter, but testing system-oriented documentation for usability would be futile. Effective usability testing of procedural discourse seeks to ascertain that the discourse "becomes an important part of the understanding process" (Albers, 2004, p. 116) for users. Studying the usability of multimodal instruction can be useful too, especially in determining the most effective mode (video or paper) for instructional discourse (Alexander, 2013).

Usability testing can enhance procedural discourse "to ensure the design contains all the features needed to invoke the proper response and that it is laid out in the manner which users expect" (Albers, 2004, p. 139). Studying how users encounter and seek answers to ill-defined problems is also an important part of a technical communicator's work (Swarts, 2018, p. 64). Content management systems complicate the process of learning about the usability of documentation (Hovde, 2019), but usability remains a crucial part of an effective process that also includes deliberate effort in the complexities of understanding users as well as other communication variables, as discussed in the next section.

Understanding Users' Ways of Learning and Their Uses of Procedural Discourse

Among the communication variables (audience/user, content, author, and format), audience may have the strongest influence in decisions about procedural discourse (Hovde, 2010). Using a rigorous process of understanding users and their needs and goals can make procedural discourse more effective (Albers, 2004; Hovde, 2001), as technical communicators benefit from rich approaches to learning about users rather than speculating about user characteristics (Hovde, 2001). They need to collect "information from a full spectrum of users so the range of knowledge and detail requirements" (Albers, 2004, p. 133) is well understood. For instance, technical communicators might follow conversations in online user forums to discover issues users typically encounter (Swarts, 2018, p. 85). Because "the writer must negotiate the flow of information from the perspective of the user" (Hogan, 2013, p. 157), understanding users is central to making decisions while creating procedural discourse (Hovde, 2010). Whatever the process of developing an understanding of users, those perceptions are crucial to developing effective procedural discourse.

Understanding that users gain procedural knowledge through a variety of modes of instruction combined with practice can lead to innovative forms of materials to help users; visual technical communication in its many forms is especially important because the brain processes visual information more immediately and strongly than linguistic information (Remley, 2015, pp. 28-29). Furthermore, including auditory instruction as it complements other modes of instruction can help users learn (Remley, 2015). Technical communicators need to consider learning theories such as "cognitive load" and "constructivism" (Hogan, 2013, p. 159) when designing task-oriented procedural discourse that adapts to users' learning capabilities. Technical communicators may also need to consider that users often simply scan written instructions rather than reading them in their entirety (Loorbach et al., 2006).

In addition, effective technical communicators need to work within a context in which they have access to information about users and content (Hovde, 2000, 2001, 2002). They need to consider if the typical purpose of the documentation is a tutorial for novices or a reference for experienced users who need to refresh their knowledge (Farkas, 1999)—or a combination of the two. Overall, technical communicators need to understand "how the user thinks and what the user needs so that interface operation, content, and presentation can maximize their respective potentials in communicating with the user" (Albers, 2004, p. x). Of all the communication variables, the user is the most challenging to understand and address in creating procedural discourse.

Technical communicators need to consider users' knowledge levels when designing user documentation. One of the decisions technical communicators need to make is how much detail, especially of conceptual knowledge, to include, whether to create streamlined or detailed documentation. These decisions may affect the ethos or the credibility that users assign to the documentation and the organization that provides it.

Technical communicators may also benefit from the insights of neuroscience as they consider how to assist users in developing procedural knowledge. Specifically, technical communicators need to understand that learning new processes involves more than cognition; it also includes practice/movement to help learn and reinforce that learning. (Remley, 2015, p. 34). Additionally, as they design learning experiences and materials, it is useful to consider the role users' prior experience plays in learning (Remley, 2015). Furthermore, technical communicators need an understanding of the following five principles that apply to how users learn new tasks:

- they prefer to integrate two or more senses as they learn,
- the visual is perceived first and often dominates,
- the timing of when information is received relative to other information affects how it is learned,
- prior experience/learning style affect how they learn, and

- users may focus on one mode of instruction more than others when learning (Remley, 2015, p. 40).

To adapt documentation to users' situations of use, technical communicators perform best when they understand that "people interact with programs differently at various times, depending on their job tasks, their professional approaches to these tasks, and the problems or breakdowns that they encounter during a task" (Mirel, 1993, pp. 25-26). Technical communicators also benefit when they realize that "documented instructions are not rote actions but are interpreted in use, succeeding only so far as that interpretation leads to improved intellective skills, coordination of social interactions and team efforts, and innovative approaches to business processes" (Mirel, 1993, p. 26). Users frequently move quickly beyond routine tasks and make inventive adaptations to suit their workplace or other contexts, especially if they are in roles that require or encourage non-routine usage of the system. These users move beyond an "automated" stance to an "informated" position (Mirel, 1993, p. 37). Users are more interested in acting within a situation, which goes beyond simply acting within a system. If procedural discourse does not include sufficient information, users need to "invent a procedure in the process of applying a tool" (Paradis, 1991, p. 269). Overall, procedural discourse "ought to address the point where user's motivation intersects with technology . . . , a picture of the technology that is inseparable from our situated uses of it" (Swarts, 2018, p. 134). In these ways, documentation can help users develop procedural knowledge.

In teaching people how to use a simple system, technical communicators benefit from task analysis of how users might employ the system. However, in providing instruction in more complex systems, creating user guidance becomes more challenging (Albers, 2004; Swarts, 2015). A simple system and a complicated system are similar in that "various problems can be plotted out and addressed" (Albers, 2004, p. 17), but a complex system is more open-ended with multiplying possibilities for use. Traditional documentation can provide a sense of stability that reflects "assumptions about use, and assumptions about principles that matter most in understanding that technology" (Swarts, 2018, p. 42). Procedural discourse may be able to provide "information with which the knowledge and skills can be built to find their own solutions to their own problems" (van Loggem, 2013, p. 172). However, users' issues may go beyond those situations.

Complex situations are not new. For example, Karen Schriver (1997) provided an example of attempting to set up two VCRs to copy and edit videotapes, a process that also involved a "cable outlet, a converter box, and a TV" (p. 228), all pieces that had to interact to help the users achieve their goals. The creators of manuals for all of these devices did not anticipate such a configuration, so the users invested many hours of trial and error trying to figure out how to make the components work together.

Technical communicators consider users' knowledge levels, needed level of detail, and their ability to process the information cognitively. Furthermore, they need to understand users' "intentions, context, knowledge, skills, and experience" (Albers, 2004, p. 68) to create useful and effective procedural discourse. At times, users may not be able to articulate their needs or their tacit processes (Albers, 2004), so technical communicators need multiple ways to understand users' needs. Overall, "audience and task analysis provides an understanding of the reader's prior knowledge, attitudes, and needs" (Albers, 2004, p. 74), allowing for appropriate design of procedural discourse.

Simply categorizing users as expert or novice may not yield a rich image of user groups and may not take into account the fact that novices may become experts. In addition, a user may be an expert in software, but not in the content relevant to larger tasks. For instance, a user may know how to use a spreadsheet to manipulate quantitative data, but may have no knowledge of principles of accounting. While this user may have an expert level of spreadsheet technical knowledge, the accounting knowledge may be at a novice level, further complicating the task of creating appropriate procedural discourse for that user.

Understanding that users' goals may shift when completing complex processes is a valuable insight for technical communicators (Albers, 2004). In addition, technical communicators creating procedural discourse for ill-defined, complex situations benefit from rigorous methods of understanding users, especially the "mental models" users may possess (Albers, 2004, p. 127), relevant to using that system. When users, especially novices, experience cognitive overload, their mental models cannot account for information, errors increase, and they may omit relevant tasks (Albers, 2004). For procedural discourse to succeed, creators need to understand users' mental models and social contexts when creating it. In addition, users under stress and time pressures in their contexts may not be able to process information well (Albers, 2004). Supplementing their complex understandings of users, technical communicators need to consider format and genre conventions as well as the affordances of various media, as explored in the next section.

Understanding Genre Conventions and Media Affordances

In addition to developing a rich understanding of users, their behaviors, their goals, and their ways of learning, technical communicators also need to understand the qualities that various media offer for procedural discourse as well as its genre conventions.

Technical communicators make decisions about the media used for technical communication, especially looking at the "affordances and constraints" of those media as they stimulate learning (Remley, 2015, p. 49). These affordances and constraints may involve the senses the medium employs to aid learning, including various combinations of auditory, visual, and/or tactile experiences. Technical communicators also need to be aware that learning is hampered when too much sensory input is included (Remley, 2015, p. 60) in whatever medium is used.

In thinking about genre conventions, it is crucial to understand that effective procedural discourse typically includes at least information about the system, an implied role for the user, a sense of the context of use, and actions the user will perform (Paradis, 1991, p. 258). Another genre convention to consider in creating procedural discourse is that including a rationale for a specific action can motivate and engage learners (Remley, 2015, p. 76).

To make information salient for users, it needs to be presented in ways that call attention to it and that make it easy to find, helping users to make sense of a situation. The most important content needs to be foregrounded, so technical communicators need to understand how people will use the system in order to understand what to emphasize (Albers, 2004). Effectively designing content provides an "adequate flow of information to the user in a form that makes sense in the situational context" (Albers, 2004, p. 83), helping users interpret meaning and achieve their goals.

Including warnings, cautions, and notes as well as other material may provide conceptual information relating to non-routine conditions. Too much conceptual detail can alienate users, but too little can leave them bereft. Formatting decisions can also help users develop procedural knowledge. However, a bit of deliberate redundancy, especially in making connections between words and visuals, may assist users in being able to understand devices and processes (Tenbrink & Maas, 2015). Visual communication can be especially crucial in procedural discourse, but it needs to be well designed for the audience and the medium (Schriver, 1997). For instance, flow charts showing conditions under which decisions need to be made may help users in non-routine situations (Farkas, 1999).

Technical communicators also need to consider the potential effects of motivational elements in procedural discourse. These motivational elements may include the roles in which the users and authors are cast; the use of non-technical terminology; the usefulness of examples, anecdotes, or metaphors; mentions of users' goals outside of the technology that may lead to specific actions; and the inclusion of testimonials. Furthermore, technical communicators may need to explore ways to balance the inclusion of these motivational elements with the conciseness and the efficacy of the instructions (Loorbach et al., 2006).

Whatever the medium and formats of the procedural discourse, technical communicators need to understand at least the affordances and the genre conventions discussed in this section in order to create effective procedural discourse, which differs in significant ways from other genres. I still encounter far too many examples of procedural discourse that try to explain steps in a paragraph format, that omit crucial visuals, and that do not pay attention to effective design of information. In addition to understanding genre conventions, technical communicators also need to consider how their discourse projects an image about the creators of that discourse.

Understanding the Relations between the Organization and Its Procedural Discourse

Because procedural discourse is often created within an organization, creators need to consider how the context influences its creation (Hovde, 2002) but also how users might perceive the organization's image based on interactions with the procedural discourse it provides. Organizational constraints and resources affect the process of creation; technical communicators may find themselves hampered when colleagues do not understand the nature or importance of effective procedural discourse (Hovde, 2002) and thus do not ensure that the technical communicators have the resources available and/or do not support an effective process for creating usable, useful procedural discourse.

In addition, well-designed procedural discourse can affect the way users perceive the organization. For instance, if an organization provides well-designed, usable, and useful online help, users are more likely to be favorably disposed to that organization. Finally, technical communicators need to think about how they understand and select content for procedural discourse.

Understanding and Selecting Content

Technical communicators ideally select and shape content appropriately so that it is adapted to user's needs. In order to do so, they need access to conceptual knowledge, such as technical specifications (Hovde, 2000), which they combine with their knowledge of the other communication variables (users, content, and organizational/authorship considerations) to create procedural discourse (Hovde, 2010). They need to select the most relevant content for users and ensure its accuracy as well. Including too much detail will overwhelm users (Salvo et al., 2007), and including too little will leave users without necessary guidance. Technical communicators without accurate and rich content knowledge may produce procedural discourse that does not meet user needs and may even lead them astray.

Although technical communicators may experience conditions that allow them to create effective procedural discourse as they consider the variables discussed in this section, several commonly held but misguided ideas may inhibit their work, as discussed below.

Myths About Procedural Discourse and Instructional Materials

Over time, I have noticed that several popular attitudes toward procedural discourse inhibit the creation of effective instructional materials. These myths need to be examined and countered when possible in order to foster the creation of procedural discourse that can empower users.

These myths include “Nobody uses instructions,” “Anyone can write instructions,” “Technical communicators are merely ‘prettying up’ technical content,” “Good procedural discourse can compensate for a system that was not designed with usability in mind,” and “Our system is well designed and intuitive, so user documentation is not needed.” This section addresses each of these misconceptions in turn.

Nobody uses instructions.

Although procedural discourse/instructional material is often vilified (Johnson, 1998) and many people believe that no one uses that material, research indicates that people do use instructional material (van Loggem, 2014), but in ways that may be unintended by the creators of that material (van Loggem, 2013). For instance, a user may ignore the instructional material initially, but then consult it after reaching an impasse in the use of the system, much as some people only consult a map after they are lost (Mirel, 1993; van Loggem, 2014). One bit of evidence that users do seek procedural discourse can be seen in the popularity of third-party publications such as the *Dummies* and *The Complete Idiot’s Guide to* . . . series aimed at teaching people to use software and complete other procedural tasks (van Loggem, 2013).

With the advent of more complex systems and open-ended tasks, users have turned to multiple means of gaining procedural knowledge that are more “interactive, quicker, and can offer more targeted assistance” (Swarts, 2018, p. 6) than traditional documentation. However, a need still exists for technical communicators who understand the dynamics of procedural discourse and how to present it effectively to users in a variety of approaches. Therefore, “If professional users of software are as willing to consult documentation as the findings suggest, then taking pains to design and develop documentation of the highest possible quality is a worthwhile endeavor” (van Loggem, 2014).

Learning to use a system via consulting written instructions is a learned behavior; “One who has learned to do new tasks through demonstration and practice and has never used a print-linguistic document will not understand how to use a manual to learn a new process” (Remley, 2015, p. 26). Hence, offering multiple modes for learning is essential.

Documentation is sometimes devalued within an organization because of the difficulties of measuring return on investment, but “in the long run, misinformed users concluding that a particular software product is useless is even more expensive” (van Loggem, 2014) than creating effective procedural discourse. The complexity of many systems implies that simply improving the interface will not be sufficient for users to learn the system (van Loggem, 2013). Hence, procedural discourse is needed, but it may also need to appear in innovative, user-centered formats.

Anyone can write instructions.

Simplifying procedural knowledge into procedural discourse may look easy, but it is actually complex (Johnson, 1998). While it is true that anyone can create some sort of instructions, not everyone can create them to be effective in achieving their goals. In fact, engineers and marketing personnel may create instructions that lead to injury and death (Paradis, 1991). Crucial skills for creating effective procedural discourse include (but are not limited to) knowing how to learn about subject matter (Hovde, 2001), knowing how to learn about users (Hovde, 2000), and knowing how to work within organizational situations to follow a productive process (Hovde, 2002). In addition, technical communicators today need to know how to use content management systems and other tools to create, manage, and distribute procedural discourse in its many forms.

Technical communicators are merely "prettying up" technical content.

This myth assumes that presentation can be separated from content, but actually, content does not exist outside of presentation. Instead, technical communicators transform descriptive material based on their knowledge of the technology, the audience, the image that their organizations wish to project, and the best means of communicating procedural discourse to the intended users (Hovde, 2010). Technical communicators select appropriate content for the users' situations of use (Paradis, 1991) rather than offering only exhaustive documentation. Presentation of complex information is crucial to users' abilities to engage with a system and understand it (Albers, 2011). Technical communicators actually serve as knowledge creators (Hovde, 2010) and knowledge managers (Swarts, 2018). Merely "prettying up" content often leads to user-friendly discourse that may be readable and engaging, but does not help users in developing active procedural knowledge.

Good procedural discourse can compensate for a system that was not designed with usability in mind.

Even after years of efforts to create usable systems, far too many systems are not designed with principles of effective human-system interaction in mind. Some system designers hold the attitude that training and user documentation can teach people to use a system that is difficult to use (Albers, 2004). However, technical communicators themselves may become frustrated with a poorly designed system and may despair over how to create effective procedural discourse for that system. Indeed, technical communicators may serve as user and/or usability advocates if they are able to participate early in the process of designing a system to be usable.

The system is well designed and intuitive, so user documentation or procedural discourse is not needed.

In contrast to the previous myth, some interface designers for years have claimed to provide "intuitive" interfaces that do not require user instruction (van Loggem, 2013). However, unless users have undergone the appropriate experiences that lead them to be able to use a system without documentation (deWinter, 2014; Paradis, 1991), creating an intuitive interface is more challenging than designers might think, especially for complex technologies because "access to more complex technologies . . . usually requires a formal framework of explanation . . . that illustrates the contexts and conditions of effective action" (Paradis, 1991, p. 264). Many users lack the mental models needed to comprehend and use a new interface, especially a complex one.

So-called "Intuitive" interface design is typically based on socially constructed experiences and direct instruction rather than the innate features of human nature. For instance, if experienced drivers rent an unfamiliar model of car, they know from years of interacting with automobiles to look for common dashboard controls—headlight switch, wiper control, ventilation controls, etc. Designers of automobile dashboards are also familiar with conventional controls and have usually placed them in accessible places. However, at times, an unfamiliar control is present. For instance, many cars now have a way to turn off "traction control" when one is stuck in mud or snow. However, if drivers are not familiar with this feature, they may not know what the button marked "TC" does and may have to consult the owner's manual, which ideally will provide them with procedural and conceptual knowledge.

"Intuitive" design is thus based on commonly shared experiences and knowledge, which lead to procedural and conceptual knowledge that help users navigate interactions with new systems. These experiences create a mental model that guides how users interact with an unfamiliar and/or complex system. A mental model, built from previous experience, "corresponds to the cognitive layout that a person uses to organize information in memory" and "helps to make connections among disparate bits of information" (Albers, 2004, p. 135). Creating effective procedural discourse benefits from a rich understanding of users' mental models that influence how they learn new information and processes.

An interface that is easy to use generally calls on conventional features and practices, but usually these interfaces are connected to relatively simple processes and systems. In addition, an easy-to-use interface may employ metaphors with which users are familiar. For instance, designers of early graphical user interfaces employed symbols for common office items such as a desktop or a trash can. Users could then take their previous knowledge and transfer it to using the interface. However, when processes and systems become complex, "intuition" may not suffice.

Procedural discourse is part of the user interface (Johnson, 1998; Suchman, 1987), mediating between the intentions of the system designers and the goals of the users, influenced by the technology itself. Because "the lay person is largely isolated from the professional origins of technologies" (Paradis, 1991, p. 257), some form of procedural discourse is necessary for effective use of complex technologies. This procedural discourse "becomes a kind of script for the human-machine interface, in which human physiology is unified with machine action to achieve a utilitarian objective . . . [that can] . . . direct the human-machine interaction so as to deliver the technology to the user's purpose" (Paradis, 1991, p. 268). However, that discourse needs to go beyond simple "how-to" knowledge to help users understand the consequences of their actions (Paradis, 1991, p. 275).

In addition, in an imperfect world, systems are not always thoughtfully created with a focus on users and usability, so user documentation is needed (van Loggem, 2013). Because some systems may need to be versatile and provide a variety of functions, they are necessarily complex. In a complex system, the interface may not be able to provide a rich view of that system to users, but documentation can assist users in understanding the system and its possible uses (van Loggem, 2013). In this sense, procedural discourse is a crucial part of the interface between users and the system (Suchman, 1987).

Because of the complexity of creating effective procedural discourse and the prevalence of the myths discussed in this section, future historical and empirical research is crucial for improving understandings of and the creation of procedural discourse.

Areas for Future Research

The nature of procedural knowledge and effective procedural discourse is worthy of further study. The following questions may guide further exploration and inquiry:

1. *What can history teach about principles of effective procedural discourse?* Although some historical research has been done for technical communication in general (Kynell & Kynell-Hunt, 2000; Schriver, 1997; Swarts, 2018), even more insights from the past would be useful for people creating procedural discourse today so that they could understand the effectiveness of a variety of approaches that have been tried over time.
2. *What are best practices for creating procedural discourse for complex processes as we move into the future?* Creating procedural discourse for routine situations is complex enough, and much about this topic has been explored. However, creating procedural discourse for complex, interlocking systems still needs further research (Albers, 2004; Swarts, 2018).
3. *What are the forces that prevent the creation of user-focused procedural discourse? How can those constraints be addressed?* Although much scholarship has focused on the qualities of effective procedural discourse, much of that discourse does not reflect best production practices or the conditions

under which technical communicators created that discourse. Technical communicators work in complex contexts with varying constraints and resources (Hovde, 2002), so future observational studies of influences on the processes of creating user documentation can provide useful insights about the contextual factors that enhance and inhibit the creation of effective procedural discourse.

4. *When and how do users experience procedural discourse?* Although several empirical studies have been completed on this topic (van Loggem, 2013, 2014; Swarts, 2018), much more work is needed to confirm and/or counteract some of the received "wisdom" about procedural discourse. This research should draw on multiple relevant disciplines such as instructional design and cognitive science, which already have rich insights about how learning occurs, so that "the informed design of software documentation demands that the choice for medium and format of the communication, as well as its content, be based on an understanding of the underlying processes of people interacting with software and with documentation" (van Loggem, 2013, p. 176). Results of this research could provide valuable guidance to technical communicators.
5. *How do cultural contexts affect how users access, interpret, and use procedural discourse? What are the effects of procedural discourse on users' access to technology?* Grounded in the current focus on social justice in technical communication (Walton *et al.*, 2019), researchers could explore how technical communication relates to "traditionally marginalized and excluded perspectives, populations, and positions" (Jones *et al.*, 2006, p. 13), including the varied ways members of cultural groups around the world create and use procedural discourse. As technology and technical communication become more globalized, research into cultural and social considerations in procedural discourse will become more crucial.

Conclusion

Procedural discourse works best when it is designed to help users create and carry out procedural knowledge in action; however, it can also provide useful conceptual knowledge to help users address non-routine, complex, and open-ended situations.

Understanding the need for procedural discourse that adapts well to users' situations and needs is central to technical communication. Although scholarship has addressed the dynamics of procedural discourse over several decades, discourse intended to assist users in gaining procedural knowledge is still far too often poorly designed and not tested to see if it meets its goals. With the complexity of technology and other systems increasing exponentially, users need procedural discourse that is well designed to assist them in developing procedural knowledge.

Although a few processes are as “easy as 1, 2, 3,” many are not (Swarts, 2018) and thus require thoughtfully created, user-task-oriented discourse in many forms. Technical communicators need to understand the differences between system knowledge/discourse and procedural knowledge/discourse. Additionally, their colleagues who influence the nature of the documentation also need this understanding as technology and its communication become increasingly complex. In addition, technical communicators and their colleagues need to understand processes that enhance the creation of effective learning experiences for users. Well-designed procedural discourse empowers users in multiple contexts as they create and employ procedural knowledge for numerous purposes.

References

Albers, Michael J. (2004). *Communication of complex information: User goals and information needs for dynamic web information*. Lawrence Erlbaum Associates.

Albers, Michael J. (2011). Design and usability: Beginner interactions with complex software. *Journal of Technical Writing and Communication*, *41*(3), 271-287.

Alexander, Kara P. (2013). The usability of print and online video instructions. *Technical Communication Quarterly*, *22*(3), 237-259.

Associated Press. (2019, June 20). Capt. “Sully” Sullenberger slams Boeing for inadequate pilot training on the troubled 737 Max. *Fortune.* http://fortune.com/2019/06/20/capt-sully-sullenberger-boeing-737/

Barnum, Carol M. (2011). *Usability testing essentials: Ready, set…test!* Elsevier.

deWinter, Jennifer. (2014). Just playing around: From procedural manuals to in-game training. In Ryan Moeller & Jennifer deWinter (Eds.), *Computer games and technical communication: Critical methods and applications at the intersection* (pp. 89-106). Routledge.

Durack, Katherine T. (1997). Gender, technology, and the history of technical communication. *Technical Communication Quarterly*, *6*(3), 249-260.

Durack, Katherine T. (1998). Authority and audience-centered writing strategies: Sexism in 19th-century sewing machine manuals. *Technical Communication*, *45*(2), 180-196.

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

Hogan, Diane B. (2013). Theories that apply to technical documentation. *Connexions: International Professional Communication Journal*, *1*(1), 155-165.

Hovde, Marjorie Rush. (2000). Tactics for building images of audience in organizational contexts: An ethnographic study of technical communicators. *Journal of Business and Technical Communication*, *14*(4), 395-444.

Hovde, Marjorie Rush. (2001). Research tactics for constructing perceptions of subject matter in organizational contexts: An ethnographic study of technical communicators. *Technical Communication Quarterly*, *10*(1), 59-95.

Hovde, Marjorie Rush. (2002). Negotiating organizational constraints: Tactics for technical communicators. *Technostyle*, *18*(1), 61-94.

Hovde, Marjorie Rush. (2010). Creating procedural discourse and knowledge for software users: Beyond translation and transmission. *Journal of Business and Technical Communication*, *24*(2), 164-205.

Hovde, Marjorie Rush. (2019). Effects of content management and organizational context on technical communication's usability. In Guiseppe Getto, Jack Labriola, & Sheryl Ruszkiewicz (Eds.), *Content strategy in technical communication* (pp. 69-88). Routledge.

Hovde, Marjorie Rush & Renguette, Corinne C. (2017). Technological literacy: A framework for teaching technical communication software tools. *Technical Communication Quarterly*, *26*(4), 395-411.

Johnson, Robert R. (1998). *User-centered technology: A rhetorical theory for computers and other mundane artifacts*. SUNY Press.

Jones, Natasha N., Moore, Kristen R., & Walton, Rebecca. (2016). Disrupting the past to disrupt the future: An antenarrative of technical communication. *Technical Communication Quarterly*, *25*(4), 211-229.

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

Kynell, Teresa C. (2000). *Writing in a milieu of utility: The move to technical communication in American engineering programs, 1850-1950* (No. 12). Greenwood Publishing Group.

Loorbach, Nichole, Steehouder, Michaël, & Teal, Erik. (2006). The effects of motivational elements in user instructions. *Journal of Business & Technical Communication*, *20*(2). 177-199.

McCloud, Scott. (2008). *Google Chrome*. https://www.google.com/googlebooks/chrome/

Mirel, Barbara. (1993). Beyond the monkey house: Audience analysis in computerized workplaces. In Rachel Spilka (Ed.), *Writing in the workplace: New research perspectives* (pp. 21-40). Southern Illinois University Press.

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

Purdue Employees Federal Credit Union. (2008). *Manage your money on the go: Set up your free online and mobile banking services today*. Purdue Employees Federal Credit Union.

Remley, Dirk. (2015). *How the brain processes multimodal technical instructions*. Baywood Publishing.

Roochnik, David. (1996). *Of art and wisdom: Plato's understanding of techne*. Penn State University Press.

Practical Peripherals. (1993). *PM14400FX MT Modem: Operating Manual.* Practical Peripherals: Thousand Oaks, CA

Salvo, Michael, Zoetewey, Meredith, & Agena, Kate. (2007). A case of exhaustive documentation: Re-centering organizations around user needs. *Technical Communication*, *54*(1), 46-57.

Schriver, Karen A. (1997). *Dynamics in document design: Creating text for readers*. Wiley Computer Publishing.

Sears, Roebuck and Company. (n.d.) *Operating suggestions and Coldspot recipes*. Sears, Roebuck, and Company.

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

Swarts, Jason. (2014). The trouble with networks: Implications for the practice of help documentation. *Journal of Technical Writing & Communication*, *44*(3), 253-275.

Swarts, Jason. (2015). Help is in the helping: An evaluation of help documentation in a networked age. *Technical Communication Quarterly*, *24*(2), 164-187.

Swarts, Jason. (2018). *Wicked, incomplete, and uncertain: User support in the wild and the role of technical communication*. Utah State University Press.

Tenbrink, Thora, & Maas, Annika. (2015). Efficiently connecting textual and visual information in operating instructions. *IEEE Transactions on Professional Communication*, *58*(4), 346-366.

van Loggem, Brigit. (2013). User documentation: The cinderella of information systems. In Álvaro Rocha, Anna M. Correia, Tom Wilson, & Karl A. Stroetmann (Eds.), *Advances in information systems and technologies*, *Vol. 206* (pp. 167-177). Springer Science and Business Media.

van Loggem, Brigit. (2014). "Nobody reads the documentation": True or not? In *Proceedings of ISIC*, *the Information Behaviour Conference, Leeds* (pp. 2-5). Information Behavior Conference.

Walton, Rebecca, Moore, Kristen, & Jones, Natasha. (2019). *Technical communication after the social justice turn: Building coalitions for action*. Routledge.