ABSTRACT

INTERNING AT CONVERGYS CORPORATION: TECHNICAL EDITING IN A TECHNICAL DOCUMENTATION TEAM

By Tyler A. Parris

This report contains a detailed exploration of my internship at Convergys Corporation, where I worked as a technical editor in the technical documentation (tech docs) team of the User Performance Support (UPS) group from June - October, 2002. In Chapter 1, I explain the history and organizational dynamics at Convergys and detail the role of the UPS group. In Chapter 2, I describe my projects and contributions to the tech docs team, and I introduce my core project, which was the creation of a style guide for the tech docs team. In Chapter 3, I detail how I used the Anderson Problem-Solving model to identify the problem that resulted in the style guide and to arrive at a suitable solution. In Chapter 4, I analyze this application of the problem-solving model, and I conclude by explaining how the MTSC program equipped me for work at Convergys and beyond.
INTERNING AT CONVERGYS CORPORATION: TECHNICAL EDITING IN A
TECHNICAL DOCUMENTATION TEAM

An Internship Report

Submitted to the Faculty of Miami University
in partial fulfillment of
the requirements for the degree of
Master of Technical and Scientific Communication
Department of English
by
Tyler A. Parris
Miami University
Oxford, OH
2004

Advisor: ____________________________
Katherine Durack

Reader: ____________________________
Paul Anderson

Reader: ____________________________
John Benamati
# TABLE OF CONTENTS

ACKNOWLEDGMENTS ........................................................................................................ III

CHAPTER 1: INTRODUCTION .............................................................................................. 1
  Convergys’s Corporate History .................................................................................. 1
  Convergys’s Corporate Culture .................................................................................. 2
  Convergys’s Organizational Structure and Technical Communication ................... 3
  The Writing Process of the Tech Docs Team .............................................................. 5

CHAPTER 2: MY PROJECTS AND CONTRIBUTIONS TO CONVERGYS ......................... 7
  Providing Comprehensive Editing and Copyediting Services .................................. 8
  Building Relationships ............................................................................................. 10
  Advocating for Technical Communication and Best Practices .............................. 12
  Looking Toward the Main Project ............................................................................ 13

CHAPTER 3: CREATING A TECH DOCS STYLE GUIDE USING THE ANDERSON
  PROBLEM-SOLVING MODEL ....................................................................................... 14
  The APS Model ......................................................................................................... 14
  Step 1: Defining the Problem .................................................................................... 15
  Step 2: Designing a Solution ..................................................................................... 17
  Step 3: Testing the Solution ...................................................................................... 20
  Step 4: Implementing the Solution ........................................................................... 20
  Omission of the Evaluation Step .............................................................................. 21

CHAPTER 4: ANALYZING THE APS MODEL AND THE APPLICATION OF MTSC
  SKILLS IN THE WORKPLACE ...................................................................................... 23
  The Difficulty with Audience and Context .............................................................. 23
  A Metamodel for Applying the APS Model ............................................................... 24
  The Difficulty with Solving Grammar and Style Problems ...................................... 25
  Application of MTSC Skills on the Tech Docs Team .............................................. 26

APPENDIX A: HANDOUT FROM A BROWN-BAG LUNCH PRESENTATION .................. 27

APPENDIX B: SAMPLE TOPICS FROM THE UPDATED UPS STYLE GUIDE .................. 30
  Language and Style ................................................................................................. 30
  Phrases, Usage, and Nomenclature .......................................................................... 31
  Format and Punctuation .......................................................................................... 34

BIBLIOGRAPHY ............................................................................................................ 36
Acknowledgments

I would like to dedicate this internship report to my classmates and faculty at Miami University, both in the MTSC program and beyond, who helped me learn and grow as a technical communicator and as a person. I would also like to thank the UPS technical documentation writers, Bob Blankenship, Connie Spaw, Julie Carter, Tom Rose, and Tony Popp; my supervisor, Peggy Girten; my manager, Tracey Tebelman; and my colleague from the user documentation team, Mary Kendig, for their roles in teaching me, listening to me, and ultimately helping me solve some significant problems for the team. Finally, I would like to thank my mother, father, and Katie for their tireless support of my academic and professional efforts, without which this report would not be possible.
Chapter 1: Introduction

To fulfill the internship requirements for my Master of Technical and Scientific Communication (MTSC) degree at Miami University, I worked for Convergys Corporation in Cincinnati, Ohio, from June 2002 to October 2002. In Convergys’s User Performance Support group, I worked as a technical editor supporting Peggy Girten, a user performance specialist who was responsible for managing technical documentation projects.

In this chapter, I describe the history and organizational dynamics at Convergys and explain the role of the User Performance Support (UPS) group within the larger organization.

Convergys’s Corporate History

Convergys Corporation, formed in 1998 when CBIS and Matrixx Marketing, Inc., spun off from Cincinnati Bell, Inc., has become the world’s largest provider of outsourced customer service. Since 1996, the company has provided customer service call centers, software, and representatives for companies across many industries. In that time, the number of Convergys employees taking customer service calls on behalf of those companies has grown from 15,000 to 41,000. At the time I wrote this report, Convergys was a major force in the global customer service market, with a total of 45,000 employees in 30 countries.

In terms of both reputation and resources, Convergys benefited from its origins at Cincinnati Bell and became a pioneer company in its own right. Recently, the company began shifting its focus. Instead of simply providing outsourced customer service call centers and representatives, it began providing its billing and customer care software to companies with in-house customer service programs. This shift has earned Convergys an even greater share of the customer service outsourcing market, as demonstrated by its growth from $400 million in revenue in 1996 to $2.2 billion in 2000.

---

Convergys’s Corporate Culture

Like its parent company, Convergys structures its people and teams hierarchically, and I would describe the organizational structure as hierarchical, centralized, and formal. Presumably, Convergys evolved into this hierarchy to reduce the complexity of the organization’s products and goals into straightforward units and tasks, which are managed and directed from top-level employees to lower-level employees. Figure 1 illustrates this hierarchy.

There was an unspoken but clear social hierarchy in the company. With few exceptions, entry-level employees and support staff worked in cubicles on the inside of the building, managers enjoyed the perquisite of cubicles by windows, and employees at the director level or higher enjoyed offices of their own, significantly more office space, more substantial desks and equipment, and windows. Lower-level employees didn’t socialize with directors, VPs, or
higher officers in the company on a day-to-day basis. If a lower-level employee wanted to communicate with those higher up, he or she did so through the chain of command.

Generally, supervisors at Convergys used a loose command-and-control model to ensure employees were on track with scheduled tasks. They emphasized employees being present and working visibly around the office as long as necessary to complete projects, and they carefully monitored work schedules to ensure that each team member contributed.

A combination of organizational and personal goals dictated which projects I worked on, how I would best approach them, and to whom I reported. I identified these goals during my pre-employment interview, during which I evaluated the internship for compatibility with MTSC program requirements. The organizational goals were negotiated by the group director, Bill, and the team manager, Tracey, who then negotiated with my immediate supervisor, Peggy, and me. During this negotiation process, I clarified with Tracey and Peggy my expectations for the UPS team: that we would meet weekly to discuss my progress, that I would be allowed to share and publish my findings, and that UPS would evaluate my work and share the evaluation with my committee. In return, the UPS representatives clarified their expectations for me: that I would perform various levels of edits on specific documents, interact regularly and respectfully with coworkers, edit approximately 800-1500 pages per week, and publish my final report only after affording Convergys an opportunity for a legal review. During this iterative and collaborative process, each of us expressed expectations, agreed to them or added new ideas as appropriate, and codified the results in the formal internship proposal.

**Convergys’s Organizational Structure and Technical Communication**

As a technical communicator, I wanted to work for Convergys because good customer service — the basis for its business — was driven by a technical communication question that I hoped to answer during the internship: to what extent can good writing and technology solve problems for customers? While I did not answer this question definitively during my internship, nor do I focus on it in this report, it was an important motivator for me in choosing the Convergys internship. After all, the people calling in to Convergys call centers were
simply users of technology, and Convergys profited from enabling people, through print and Portable Document Format (PDF) technical manuals, to use that technology.

Specifically, Convergys’s users needed quick access to useful information about their own customers. Convergys developed software to meet this need and marketed the software with its existing services to companies who used combinations of software and services as part of their own in-house customer service efforts. For example, one tool that Convergys created to help its clients was Atlys®, a billing and customer care software application and the flagship product for Convergys in the wireless telecommunications industry. Atlys provided real-time processing of customer data throughout the billing and customer care process for wireless providers. The customer care process included the mediation, activation, rating, and billing of wireless services. In other words, cellular service providers and their customer service representatives could use Atlys to negotiate plans and billing rates with customers (mediation), activate customers’ cell phone services (activation), measure the customers’ usage of the services (rating), and bill customers for the services used (billing). Without effective documentation, a client might have difficulty supporting those who used this complex application.

The team I worked for played an important role in the overall success of the UPS group — and ultimately Convergys — because it ensured the production and quality of Atlys documentation and because it began to influence product design with Atlys developers.

To produce good documentation within the company’s resources, the UPS group combined human factors specialists, writers, and editors across two primary documentation teams, the technical documentation (tech docs) team and the user documentation (user docs) team. Both teams primarily documented software after it had been developed. The tech docs team documented tasks for the system administrators who would install, administer, and troubleshoot Atlys. The user docs team documented tasks performed by the customer service representatives using Atlys software to interact with customers. I edited documents for the tech docs team, supporting six technical writers under Peggy, who had served as the team’s editor and supervisor during the previous year.
The Writing Process of the Tech Docs Team

Technical writers in the tech docs team played a significant role in shaping the content of technical documentation. They worked with subject matter experts (SMEs), through a carefully designed process, to turn detailed technical knowledge into readable and usable documentation. As the team’s only dedicated editor, I assumed most editorial responsibilities within this process. My responsibilities included a range of tasks, from making detailed copyedits to recommending new processes. Figure 2 illustrates the documentation development, distribution, and post distribution process of the tech docs team. I performed my editorial tasks during the ALPHA Development, BETA Development, and Source Document Release phases, in support of those writers who started and completed the process.
I will use the remainder of this report to detail my short-term role in the organization’s long-term success. In Chapter 2, I describe my projects and contributions to the tech docs team and UPS, and I introduce my core project. In Chapter 3, I describe my primary project in detail, including how I used a problem-solving model to identify a significant problem in the processes of the UPS team and arrive at a suitable solution. In Chapter 4, I analyze the application of the Anderson problem-solving model to my project, and I conclude by explaining the ways the MTSC program equipped me for work at Convergys and beyond.
Chapter 2: My Projects and Contributions to Convergys

Peggy and Tracey made it clear at the beginning of the internship that they had hired me for my organization, grammar, and proofreading skills and that they would afford me a modest amount of autonomy in the day-to-day performance of my job. They would prioritize documents and determine which ones I would edit and when. Peggy would meet with me once a week to provide feedback and direction and to field questions from me. At the same time, my word on editing matters would stand as authoritative, and I would be free to initiate projects in addition to my day-to-day editing as the need arose.

As a result of this autonomy, I established some goals at the beginning of the internship that would help me get the most out of my internship: I wanted to provide comprehensive and copy editing to the team, develop and maintain positive editor-author relationships, and advocate technical communication within the larger organization at Convergys. Although I worked on a number of projects on my way to reaching those goals, I primarily copyedited documents on a day-to-day basis for the tech docs team. Additionally, I prepared and delivered a brown-bag lunch presentation for writers on the team. I also wrote a memo to Peggy about the need for both a user-centered approach in the team’s documentation and a company-wide approach to documentation. Figure 3 illustrates the time I spent working on editing guides, creating and maintaining a style guide, attending staff meetings, and other tasks associated with my internship, such as meeting informally with writers and e-mailing team members.
In my first five weeks at Convergys, I copyedited nearly 4,500 pages in 21 different guides, performed comprehensive editing on 200 additional pages, and checked against a list of requirements (checklist edited) nearly all of those pages. Comprehensive editing included editing for organization, structure, high-level content, and context\(^2\). Copyediting at Convergys involved nearly every other type of edit, from completeness and accuracy to sentence- and word-level revisions. Checklist editing meant checking documents against a peer editing checklist that the tech docs team had developed, which listed specific items to be double-checked before a document could be considered complete. Additionally, some documents required a special edit for very specific issues. For example, in the 840-page Billing and Customer Care Application Programming Interface Guide (BCC API Guide), I edited only content that Tom — a writer — had recently added to the guide. This edit pass ensured the consistency of added content with existing content.

I knew from my negotiation of the internship proposal that the tech docs team expected me to edit a significant page count in a short amount of time, so I quickly learned to prioritize my work. Unless a document came to me specifically for a copyedit pass, I focused my energy on solving organization, structure, and grammar problems instead of punctuation and capitalization problems. I focused this way because the writing was generally strong, and most punctuation and capitalization changes I made or recommended were minor. In my experience, such minor problems tend not to cause the same amount of confusion for readers that larger, organization problems do.

As I performed copyedits, checklist edits, and more specific edits over the first few weeks, I found a number of inconsistencies within and among documents. For example, certain words were often treated differently, even when used in the same context: setup (one word) and set up (two words), Web and web, email and e-mail, zip code and ZIP code. Also, acronyms often were not spelled out on the first mention. Even though common terms like DBA (database administrator) could arguably be appropriate for our audience without spelling them out, others were not always so obvious, like CRS (consolidated roaming system).

In addition to noting inconsistencies in the documents, I noted the need for both high-level and sentence-level changes. For example, I questioned the necessity of having a “Before you begin” section as a preface to all the guides. My own experience with studying, using, and writing manuals had indicated that people tend to skip such introductory sections and try to find the specific tasks they seek to accomplish. Peggy explained to me that we should not delete “Before you begin,” because UPS had designed its guides to be read from front to back, with all steps being followed in sequence. I addressed this concern more fully in an email memo, which I will explore later in this chapter. In other comments that I made about high-level issues, I recommended parallel headings; consistency of terminology, usage, and formatting; the use of the second person where possible; the standardization of procedural steps and their introductions; and the appropriate application of trademark and registered trademark symbols to Convergys and third-party product names. At the sentence level, I restructured as many sentences as possible so that they followed a subject-verb-object pattern or began with the structural core of the sentence. I simply reworded some sentences for clarity and word economy.
The documents I copyedited contained enough editing challenges that I easily met my first goal of providing comprehensive and copyediting services to the team. In fact, based on records that I kept from June 20 to August 8, I spent most of my time on the tech docs team providing these copyediting services. Figure 4 illustrates the time I spent working on specific editing tasks during my internship.

![Figure 4. Hours per editing task chart: the amount of time spent on each of four editing tasks from June 20-August 8. Percentages are rounded to the nearest whole number.](image)

**Building Relationships**

When I started my internship, I detected a mistrust of editors among the writers and I set out to build positive editor-author relationships. Not knowing the basis of this seeming mistrust, and thinking that I probably would not uncover all of the reasons for it in my short time at Convergys, I developed a short-term plan for building positive relations with the writers. Even if I were just out of grad school, I wanted show each writer that I was a competent and enthusiastic editor who could improve their everyday experience of writing and add value to their product. I invited each of the writers to lunch during the first couple of weeks so that we could learn about each other, find out what our motivations were, and talk about our mutual expectations. One writer was proud of his Scottish heritage, another was happy that she was getting married over the summer, and yet another was pursuing a position
on a town council. Getting to know the writers I worked with was the first step toward good working relationships. This step built trust and respect from the beginning of my internship and seemed to minimize the mistrust that I had first detected.

For the most part, writers received my editorial comments well, although sometimes, writers would question me about comments or changes. At those times, either I practiced the art of diplomacy and persuasive communication, or I admitted that I had made mistakes. As I will demonstrate later in this chapter, being diplomatic and willing to admit mistakes earned me the respect of the writers I supported. They said so in spoken conversations and in emails to my supervisors. I appreciated the good communication that resulted from my relationships with the writers because the writers helped me discover and solve problems.

For example, the inconsistencies that I found in the documentation often occurred sporadically over hundreds of pages. As a result, the writers – Bob, Connie, Julie, Tom, and Tony – discovered that my editorial comments and changes were not always consistent, and they provided me with valuable feedback about my editorial inconsistencies. I used this feedback from the writers to identify the need for a style guide, which I explore in more detail in Chapter 3.

As I edited documents over the first few weeks, the writers began asking the question, “Exactly what are you editors (Peggy and me) looking for?” The writers wanted explanations of some of the edits that I made in their documents. For example, I often changed instances of “that” to “which” and vice versa in their documents, but some writers weren’t familiar with the rules of grammar behind these changes. Some writers also did not understand why I had made other changes. Peggy suggested that I answer these questions by presenting the topic “What are editors looking for?” at a brown-bag lunch.

When preparing the topic for the brownbag lunch, I did not want to reiterate the checklists. Most of the writers held college degrees in literature, creative writing, or computer-related fields, and to have reiterated checklists would have insulted their intelligence. Instead, I reviewed the hard copy edits I had performed and talked with the writers to identify ten issues most important to them. From those issues, I created a handout (see Appendix A) explaining each issue in detail. Some of the grammatical issues were issues
that might be overlooked in common speech or in some fiction and poetry, so I added to the handout a paragraph about my philosophical perspective on technical communication and how technical communication is distinct from other forms of communication. Primarily, I explained that one goal of technical communication is to minimize a reader’s interpretation so that the communication can be translated into some kind of action, as opposed to creative writing, which encourages interpretation and is not necessarily designed to become an action.

I presented my top-ten list to the team at the brown-bag lunch, item by item. I presented each item, and then I asked for questions and comments. After we had all thoroughly discussed each topic, I moved on to the next one, until I had presented all of the topics and we had discussed them as a team. The team received the brown-bag presentation well, as evidenced by an email that Tom wrote to Tracey:

Hi Tracey. I thought Tyler did a great job today. Explaining to a bunch of writers what common mistakes they are making can easily turn into an adversarial situation. His matter-of-fact presentation with examples steered well clear of any negative connotations and, instead, turned into a positive experience. He’s proving to be a great asset to our department.3

By presenting information at the brown-bag lunch, I achieved the objective of answering writers’ questions, which helped improve writer productivity and efficiency, and I continued working toward the goal of building and maintaining positive editor-author relationships.

**Advocating for Technical Communication and Best Practices**

Sometimes, building relationships and being diplomatic required more work than did the brown-bag luncheon. During a mid-summer meeting, several team members expressed frustration at user feedback indicating that users weren’t reading the documentation provided by the tech docs team. As a result, users were having problems using certain software features and claimed those features didn’t work. We all knew that this claim was not true. The features worked well. We simply needed to find a better way to explain to users how to perform these troublesome tasks. In an email memo to Peggy, I suggested that UPS’s current documentation was designed in such a way that users had to read entire guides from front to back, often

---

3 Tom Rose, e-mail to Tracey Tebelman, 21 August 2002
before ever beginning any actual installation, operation, or maintenance tasks. I suggested this approach was not user-centered because it did not account for the task-based behavior of users. I also suggested that UPS needed to find a way to analyze tasks that users were trying to accomplish and translate that analysis into better documentation. Finally, I suggested that earlier and more frequent user testing would reduce user frustration and the costs of Convergys supporting frustrated users.

Peggy answered by explaining that she shared many of those concerns. She also said that UPS as an organization had realized many of the same things and was in the process of addressing those concerns. Peggy also suggested that Convergys, not just UPS, needed to re-think its entire documentation process. She explained that a group of people was beginning to address these issues, but the situation would not change overnight. For more information about how Convergys addressed these concerns, see “Omission of the Evaluation Step” in Chapter 4.

Looking Toward the Main Project

In the midst of all these projects, the need to solve a particular problem became evident. We needed to establish style guidelines that would guide writers and editors alike, introduce consistency in writing and editorial comments, and help minimize the time spent by writers and editors negotiating interpretation of editorial comments. This became the main problem I solved during my internship, and I will spend the next chapter explaining this problem and how I arrived at a suitable solution.

---

4 Girten, Peggy. Email to Tyler Parris. 12 July 2002.
Chapter 3: Creating a Tech Docs Style Guide Using the Anderson Problem-Solving Model

Shortly after joining the tech docs team, I quickly discovered the need for a tech docs team style guide. Creating such a style guide became my main project because it posed a significant challenge. The complexity and ambiguity of the project intimidated me at first. This project would involve gathering information from many sources, sorting that information by relevance and necessity, finalizing the format of the information, and creating a suitable deliverable in a matter of weeks. I applied Paul V. Anderson’s problem-solving (APS) model to order the complex tasks that I needed to accomplish for the style guide to become a reality.

The APS Model

The APS model is a problem-solving heuristic designed to give technical communicators a tool for thinking about and solving communication problems. Figure 5 illustrates the five steps in the APS model.

Because my internship ended at the same time that I delivered a style guide to the team, I was unable to complete step 5 by evaluating the solution. I discuss the omission of this step later in this chapter.

---

Step 1: Defining the problem

Some of the inconsistencies I found in the documentation were sporadic, sometimes over hundreds of pages. For example, if I came across the term *e-mail* on a page where the term *email* (no hyphen) was also used several times, I would change *e-mail* to match the most common usage, *email*, on that page or surrounding pages. I assumed the most common usage was probably the correct one. Many pages later, or even in another document (days later), I might find another instance of *e-mail*; only, this time, *e-mail* (with a hyphen) would be the most common usage and not *email*. There was no consistent guideline in place for determining whether *email* or *e-mail* was correct. Therefore, I made local decisions to treat terms a certain way, which led to the introduction of inconsistencies across documents. Writers and editors needed a style guide to introduce consistency, both in writing and in editorial comments, and to help minimize the time spent by writers and editors negotiating the
interpretation of editorial comments. Having style guidelines for common terms would also enable writers and editors to use the search and replace feature of Microsoft Word to ensure that every instance of an incorrectly used term could be identified and corrected.

After I identified the need for a style guide, I set out to determine the constraints and conventions within which I would design a solution. The primary constraints I faced were time and the different understandings that the writers and I had of technical communication. I needed to research the need for a style guide more thoroughly and not only develop a solution in the remaining time of my internship but do so concurrently with documentation edits. Only when I began my audience analysis, which I describe later in this chapter, did I determine the conventions I would use. I decided to rely on another team’s style guideline as the basis for ours while respecting the flexibility of my team and their need for treating certain style issues differently than other teams.

Within those constraints and conventions, I defined the audience. I saw writers and editors on the tech docs team as my primary audience. I saw SMEs, who reviewed the documentation and signed off on it, as a very important secondary audience. SMEs and writers without formal training in technical communication, working under tight deadlines, would want to know not just how to apply style guidelines but why they should follow each guideline. I had to build editorial authority into the guide, as much as possible. I would need to provide with each topic not only the most appropriate course of action for each item in the guide but also a brief justification. The inclusion of a justification would not just serve as a way for editors to prove their points to writers but would give writers and editors a tool for making intelligent style decisions. After all, style guidelines are guidelines, not absolute rules. They enable writers and editors to use language flexibly within certain agreed-upon parameters to communicate what they need to communicate.

In the process of identifying my audience, I discovered that the user docs team had been using a style guide all along. Their guide consisted of a .chm Help file, created in RoboHelp, that borrowed heavily from the Microsoft Manual of Style for Technical Publications (MSTP). The user docs style guide expanded many MSTP definitions to fit the needs of their team. Feedback I received from the user docs team indicated that the .chm format worked
well for them because guidelines were only a couple of mouse clicks away, could be stored easily on a computer’s Desktop, and did not require the production costs of print material. Because this medium worked for writers and editors, and because I wanted my style guide to maintain consistency with the existing guide, I decided to create my guide in a .chm format for the tech docs team. I ruled out a company-wide style guide because my managers wanted me to focus on UPS and leave the company-wide changes to the team that had been created for this purpose.

**Step 2: Designing a Solution**

Rather than create a separate style guide, it seemed to make more sense to start with the user docs team style guide and simply expand on existing topics and add new topics that were specific to the tech docs team. For example, certain terms, like *Cross-references*, were treated differently by the user docs team and the tech docs team and needed to have two different treatments listed under the same topic. Figures 6 and 7 illustrate the different treatments of the term *Cross-references.*
Figure 6. Cross-references topic for User Docs team. Focus is on the verbs “see,” “to go,” “continue,” and “return.”

Figure 7. Cross-references topic for the tech docs team.
The tech docs team used “Refer to” when cross-referencing other documentation, while the user docs team did not have a standard for referring to other documentation. The tech docs team also agreed that saying “Continue to” was not necessary when referring readers to the next step in a procedure. The style guide needed to account for these differences.

As the medium and format of the style guide took shape in the design phase, I developed an informal project plan for gathering information, sorting the information, finalizing the format, and creating the deliverable.

From the hard copy edits I had made and from lists sent to me by the tech docs writers, I created a list of every topic that I thought needed an explanation in the style guide. To best incorporate each topic, I simply read through the existing style guide from beginning to end and looked for topics on my list that matched. I eliminated several topics from my list because the existing style guide already addressed them. At the same time, I noted ways I wanted to expand or reword some of the current guidelines for clarity. I categorized any remaining topics under the style guide’s existing categories: Language and style; Phrases, usage, and nomenclature; and Format and punctuation. After determining which topics went under each category, I began incorporating each topic, using RoboHelp, into a copy of the .chm file.

The act of incorporating each topic meant writing clearly and briefly the guideline itself and the reasoning behind it. In some cases, I met with SMEs and other writers to clarify terms, such as the often-confused function and API, or to negotiate styles, such as whether to use a colon or a comma after for example. Incorporating the topics, as part of designing the solution, proved to be the most time-consuming task in creating the style guide. Figure 8 illustrates the time I spent working on various tasks of the problem-solving model while I developed the style guide.
Step 3: Testing the Solution

After I incorporated the topics, I solicited feedback from writers on the tech docs team. This feedback served as my user testing, because the writers were unable to make time for more in-depth testing. I printed off hard copies of all the topics I had created and defined. Then, knowing that no writer would have time to read the entire style guide, I divided the topics up among some of the writers. After receiving their feedback, I noted that some of the topics did not adequately address concerns that the writers had. For example, Tony questioned why my topic for Press instructed, “Refer to pressing a key as “Press the ENTER key,” and not simply “Press ENTER.” After gathering feedback from a number of writers and seeing how the MSTP handled the term, I agreed with Tony and changed the topic so that it did not include the and key but followed the shorter form: Press ENTER. After I evaluated the feedback from the writers, I began to implement the style guide.

Step 4: Implementing the solution

Implementing the style guide meant developing a finished draft. First, I made changes based on feedback from the writers to the topics I had created. I also emailed and telephoned the user docs team editor, Mary, who worked in Orlando, Florida, to clarify some of the
existing topics. For example, in the existing *Gerund* topic, all the examples of gerunds (nouns ending in *ing*) were incorrect: they were participles (verbs ending in *ing*). I recommended correcting this inaccuracy by finding some real examples of gerunds in the documentation. Also, the *That, Which, Who* topic needed deep revision. It originally contained a brief explanation of *that*, *which*, and *who*, and it explained that omitting *that*, wherever possible, was desirable. However, I recommended using the topic to explain restrictive and nonrestrictive clauses, which are primarily signified by the words *that* and *which*. I especially wanted to emphasize the correct use of *that* because the omission of *that* in technical documentation often leads to problems translating given passages from English into other languages. Mary agreed with most of these changes and helped me clarify other topics as well. Once I incorporated the changes based on writer feedback and my email discussions with Mary, I took the time to format each topic consistently with other topics in its category.

The last step I took in implementing the style guide was writing an email to Peggy and Mary, in which I suggested a process for keeping the style guide current. Presumably, more topics needing to be addressed in the guide would surface after I had left the group at the conclusion of my internship. I recommended the continual upgrade and maintenance of the style guide so that new issues could be addressed as needed and so that my upgrade of the guide would not be a one-time event. I emphasized the benefits of updating the guide versus the costs of not updating it, and I left the style guide and its processes, ultimately, up to the best judgment of Peggy and Mary.

I attached the style guide .chm file to that email (see Appendix B), and I left the style guide in the hands of those who own it as my final step in using the APS model.

**Omission of the Evaluation Step**

I initially omitted the Evaluate the Solution step of the APS model because my internship ended simultaneously with my delivery of the style guide, and a formal evaluation of the solution was no longer possible. Since my internship ended, however, I have learned that the UPS teams did not implement the .chm file I worked on as their primary style guide. Instead, they used my work in a much broader capacity.
The Style and Structure Committee at Convergys, a committee on which Peggy sits, formed shortly after my internship ended to evaluate product documentation across all product groups at Convergys. Peggy used the work I provided to begin negotiating company-wide standards with documentation specialists from other product groups. As Peggy writes in an email, “Probably 75% of our styles remain intact, so (your) style work was valued as the basis for a revised corporate product documentation style guide rather than (being) focused on one product…” Using Peggy’s qualitative analysis of my work, I have concluded that my solution was successful not just for the tech docs team but for the whole company.6

I will use the next chapter to analyze the APS model’s strengths and limitations as applied to my internship. I will also explain, specifically, the importance of the technical editor’s role at Convergys.

---

6 Girten, Peggy. Email to Tyler Parris. 2 April 2003.
Chapter 4: Analyzing the APS Model and the Application of MTSC Skills in the Workplace

The APS model enabled me to take an orderly approach to facing ambiguous situations. In that way, it was very useful. I noted one area in which the model could be improved, and I modified it slightly to better suit the specific way I think and work. I will use this chapter to analyze one difficulty posed by the APS model, explain the modification I made to the APS model, and conclude with the ways I applied skills that I learned in the MTSC program to add value to Convergys’s technical documentation.

The Difficulty with Audience and Context

Using Step 1 of the APS model, Identifying the Problem, I started off thinking of audience and context separately, since the model treats them as their own entities and does not emphasize that these items should be determined in any specific order. According to the model, as a technical communicator, I was supposed to identify conventions, constraints, and my audience. Yet, I soon discovered in my style guide project that I couldn’t define audience independently of the context.

As I began thinking about my primary audience members and who they were, I found myself repeatedly defining them in terms of the tasks they needed to perform and which tasks they could not perform or perform well. In other words, I defined them in terms of the problem itself. I also defined them in terms of the constraints within which they worked and the needs they had in common with one another. They seemed to have some traits that could be seen independently of their involvement in this problem, such as their demographic information — writers varied in age, formal education, experiences, and gender. But, the most useful characteristics that I discovered about my audience were their common, work-related situations. They all:

• worked under tight deadlines.

• worked within the company’s structure and culture.

• had very little formal training in technical communication.
• shared the need to communicate with SMEs (the secondary audience) quickly and persuasively.

Only after I defined the context of the situation in which team writers worked could I make what I believed to be an accurate and useful determination of who my audience was. My definition of audience was dependent on my definition of the context in which the audience worked. Therefore, it was not sufficient to present the context and audience as distinct entities, but rather I presented context as a subset of the entity called audience.

A Metamodel for Applying the APS Model

The APS model enabled me to follow an established path through the frenetic pace of day-to-day edits and the inconsistencies that the team discovered in our collective work. From that established path, I could act and react as necessary to the particular demands of the tech docs team and to the way I think and work. Even though the model worked well on my projects, I modified the model in one important way. I added a step, and I call it a metamodel for applying the model itself.

The first step in the APS model is identifying the problem to be solved. Yet, my main problem was recognizing when to formally apply the APS model as I began identifying technical communication problems within the tech docs team. The APS model seems to assume that a user will recognize the exact moment in a given situation when applying the model will be most appropriate. Put another way, if I assume that the APS model was useful for ordering my work in a chaotic work environment, then in years past I’ve approached my work using some other, probably less orderly, means. In shifting from a problem-solving approach without the model to an approach with the model, I needed to begin to recognize the precise moment when applying the model was most useful. The need to recognize when to apply the model may have seemed too obvious to include in the APS model. Yet, my inability to effectively recognize when to apply the APS model had been my most significant and frustrating obstacle in working with the problem-solving process during my MTSC coursework. It seems that whether in individual work or group projects, I received assignments and set out in a flurry of work to do them. Often, it wasn’t until I was well into a
project when I recognized the need for a problem-solving model. Then, I typically only used the model to identify and solve the main problem I was facing and not smaller problems that manifested themselves along the way.

To help me apply the APS model at more opportune times, I added a step to the APS model. The new step, Identify When to Apply the APS Model, replaces Define the Problem as the first step. It consists of a set of key words and phrases, which serve as cues for a technical communicator working in a chaotic or ambiguous work situation that it’s time to apply the APS model. For example, if I say to myself or hear in a meeting the following words and phrases, it may be a good time to apply the APS model:

<table>
<thead>
<tr>
<th>I wish we had a…</th>
<th>We need a…</th>
<th>There must be a way to…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who’s the best person for…</td>
<td>How do I/we…</td>
<td>What’s the relationship between…</td>
</tr>
<tr>
<td>Why can’t we…</td>
<td>Have we tried…</td>
<td>Is this an (adjective) idea/issue?</td>
</tr>
<tr>
<td>Can you find out how…</td>
<td>What’s the best way to…</td>
<td>When will we find time to…</td>
</tr>
</tbody>
</table>

By adding this step early in my internship, and by memorizing many of the questions so that I could recognize them in meetings, I was able to apply the APS model to my major project from its earliest stage. Otherwise, I might have applied it only after I had already completed a lot of problem-solving tasks by trial-and-error.

The Difficulty with Solving Grammar and Style Problems

When I started my internship, I had not anticipated the number of ways that terms, styles, and rules could conflict with one another, and I had not anticipated that the resolution of these conflicts would simply require editorial judgment. My confidence in editing stemmed mostly from my memorization of grammar rules and my reliance on grammar references and style guidelines. Furthermore, I thought of editorial judgment as a tool that I could use for solving the organization and structure issues of high-level editing, not for solving punctuation and style issues of low-level editing. However, writers indicated to me on several occasions that the resolution of particular grammar and style problems was more complicated than knowing the applicable grammar rule or looking at a style guide or grammar reference. For example, writers and editors applied the Letter Gothic font to utility names like `mpstat utility`. Tony pointed out that utility names can also be run from a command line and that writers and
editors applied the bold style to command line instructions, not Letter Gothic. So, in this situation, we all knew the rules, but the writers still looked to me for leadership in making a decision about how to handle this conflict in the guidelines.

I met these challenges by saying something like, “It doesn’t matter which style we use in this situation, as long as we decide on something and use it consistently in other situations like this one. Let’s use bold in this situation” (or whatever style was appropriate). The writers and my manager all seemed to appreciate the way I responded to this challenge.

**Application of MTSC Skills on the Tech Docs Team**

From the beginning of my internship, I added value to the tech docs team by applying skills I learned in the MTSC program. My ability to find grammar and style errors in text, my knack for making tactful suggestions and influencing the writers’ perceptions of editors, and my adaptation of the APS model to suit my work situation added significant value to the team and created a favorable impression within the tech docs team of the MTSC program. This value is evident in positive feedback I received from employees at several levels in the company, from the writers on my team to the director of UPS.

Specifically, I applied the art of diplomacy when Tony and I encountered a SME who insisted that most, if not all, of our suggested changes were incorrect. For example, the SME believed that neither a comma nor a hyphen were appropriate for the phrase, “small, subscriber-based systems.” The SME thought they “seem intrusive,” no matter how many references from grammar books that we used to strengthen our position in favor of both the comma and the hyphen. At first, both Tony and I were frustrated by what we saw as stubbornness and arrogance on the part of the SME. After several unsuccessful attempts to persuade the SME, I suggested a wording for the sentence that required neither a comma nor a hyphen and that did not introduce a technical inaccuracy: “small systems or systems that might be used for testing.” The SME accepted this rewrite, and the project continued.

In fact, by applying the professional skill, diplomacy, and problem-solving ability that I learned in the MTSC program, I helped my team managers and SMEs understand more fully the value of technical editing and, ultimately, technical communication in the CSD business group.
Appendix A: Handout From A Brown-Bag Lunch Presentation

What are editors looking for?  
A Quick Guide

Creative writing versus technical writing
If you’re like me, you’ve come to technical writing as an English major. You’ve studied literature or creative writing. In those disciplines, writers and readers encourage interpretation. Conciseness, clarity, and logical structure may be, but are not necessarily, values in those disciplines. What separates technical writing from those disciplines is the need to minimize interpretation. Technical writing is meant to help people accomplish tasks in often demanding environments.

Correct use of passive and active voice
I don’t care what you’re English professor taught you: passive is not always wrong. It’s more often than not inappropriate, but not always. You simply need to know when it’s useful or not. Active is preferred in most cases, because it establishes causality. Especially in technical writing, it is important for readers to understand the concepts, processes, and tasks we write about, as well as their interrelationships. In our situation, causality is important to reader comprehension. For example: “The xyz command initiates the process.” Sometimes, it doesn’t matter what causes something: “If you do x, an error message appears.” Who cares who or what displays the message? We might not even know the who or what. The point is that our reader knows not to do x. Passive is fine here.

Relative Pronouns
If I ever say something like, “It is only my love for words which makes my day complete” or “I’m doing this for the millions of people that read our documents,” I want you to roll up a newspaper and wollop me on my noggin. Which and that refer to things, who refers to people. That and which even have special, related comma usage:

- Restrictive clause (that/no comma): limits the noun’s meaning and is necessary to the sentence. For example:
  We recommend that you review the Billing and Customer Care Overview Guide, Billing and Customer Care Site Planning Guide and Billing and Customer Care Operations, Administration, and Maintenance Guide to learn more about the system, the pre-installation activities, and the planning that must take place before you install Billing and Customer Care, Release 9.1.

- Nonrestrictive clause (which/comma): provides additional information to the noun, but it does not restrict the meaning. For example:
  You will need to configure $ADMIN_DIR, which should be specified as the location of the admin directories.
**Hyphenation**
I’m not anti-hyphenation or anything. I just think people over-use these ultra-helpful little devices. A hyphen serves to establish whether a word is used as a noun, verb, or adjective. For example, “The end user is our audience” (noun) versus “End-user documentation is our business” (adjective).

**Commas separate items in a series**
Semicolons separate items in a list where any related items in the list have commas separating them already.

**Cross-references**
These need to be consistently worded: “See the “x” section on page y of z guide.”

**Sentence structure**
This is for the future: place the subject very near the beginning of the sentence: long and convoluted sentences make meaning ambiguous. There are a lot of these sentences in the current documentation. Make sure the sentence is accurately saying what it is trying to say.

For example, “If the same sequence wants to be written with the fully qualified path, then it can be written in the following two ways:” vs. “If you want to write the fully qualified path, you can write it in either of the following two ways:”

**Semantic consistency**
Use the same term to refer to the same thing every time. In one page, I’ve seen the same thing referred to in three different ways, such as:

- The SCPClient.properties file
- The SCPClient.properties
- SCPClient.properties.

This is very confusing.

Consistency can be taken too far, like the editor who once noticed the phrase, “for political, theological, and economic reasons” and who, to make the items in the list consistent (or parallel in editing jargon), rewrote the sentence “for political, theological, and economical reasons.” Yet, reasons are not “economical” (meaning they cost less): they are “economic” (resulting from conditions in the economy), so consistency actually introduced an inaccuracy here, so use judgment.

**Capitalization**
Some examples of unnecessary capitalization from one document:
“The transaction is identified by the Data Type and Service Order Type.”
“You have to select one Command Map from the list box.”
In each of these cases, the capitalized words not beginning the sentences are generic usages and have no need of capitalization. They are not proper nouns, used in a special or unusual way, and do not meet any other criteria for capitalization.

**Use of Information/Warning/Critical stop icons**
Here is an example of text from a recent guide that is inappropriately labeled as an informational note:
“This is an important table to configure. Before configuring it, you should know which table belongs in which domain. If not, the data will be transferred to the incorrect tables in the RAS domain and there is a good chance of corrupting the table.”

Since this text clearly indicates that an undesirable change or condition could occur, and implies that data could be lost, this should be at least a warning, possibly a critical stop.

Also, explain to users what the consequence is of a stated user action. It’s not enough to say, “Don’t do it.” Explain why, as in the above example, if at all possible, so that users understand the risk involved in their actions.

**Misplaced modifiers**
Modifiers (adjectives, adverbs, modifying phrases) must clearly relate to the words they modify; if they are separated, the relationship often becomes unclear. Think about these examples:

- The system was configured with just version 9.1.
- The system was just configured with version 9.1.
- Just the system was configured with version 9.1.

The placement of the modifier matters in each case.
Appendix B: Sample Topics from the Updated UPS Style Guide

The sample topics included here are taken from each of the three sections of the style guide: Language and Style; Phrases, Usage, and Nomenclature; and Format and Punctuation

Language and Style

The topics *Gerunds* and *Participles* explain and demonstrate the differences between gerunds and participles:
Phrases, Usage, and Nomenclature

The Backup (Back Up) topic explains and demonstrates the reasoning for the same term being used as one or two words, depending on its context. The same was true for NA.

Backup (Back up)

For user docs and tech docs

Use one word for the noun and adjective form of this term. Use two words for the verb form.

Example: Perform a system backup daily.

Example: Back up the system daily.

N/A (vs. NA)

For user docs and tech docs

Use N/A instead of NA when meaning not applicable, since NA may be more commonly mistaken for national association, no account, North America, or not available.
The *Dunning Notice* topic represents a topic that appeared in documentation about customer rating and billing, and nobody seemed to know how to treat it or what its origins were. The same thing was true of *ZIP code*. The standardizing of these topics required some research.
The *Glossary Entries* topic helped writers and editors create concise definitions for technology-specific terms in UPS documentation.

Glossary entries should be brief. Use fragments to save space and time, like a dictionary definition. If a glossary entry is an abbreviation, use the abbreviation as the entry heading, spell it out in the first line of the definition, followed by a period. Then, provide brief definition and end with a period, even if entry ends with a fragment. For example:

**CRS**

Consolidated Roaming System. Convergys system that interfaces with clearinghouses and Mediation Manager to process CIBER roaming records.
Format and Punctuation

The *Bold* topic explained and demonstrated how a style could be treated differently by the tech docs team and the user docs team.

**Bold**

**For user docs**

Use bold type for all system interface text (such as menu options, text labels, and command buttons).

Example: Press **Account functions**, and select **Assign a CSA**.

Contrary to regular usage, do not apply bold to any punctuation following the bold-faced term (see the above example).

**Do Not Use** bold type for window titles in text.

**For tech docs**

Use for button names, screen names, database field names, and GUI field names in procedural steps. Do not use for button names, screen names, database field names, and GUI field names in body text.

Also see [Conventions and Formats](#).
The *Commas* topic clarified one of the most frequently negotiated and confusing issues that writers and editors faced.

**Commas**

*For user docs and tech docs*

Use a comma in the following instances:

- To indicate a nonrestrictive clause. **Use:** The default option is `choose`, which should also be the default for Atlys Billing and Customer Care.

- To separate items in a list or series, especially before the last item. **Use:** After a cycle billing, the `BILL`, `CHARGE`, `CREDIT` ITEM, and `TAX_DETAIL` tables should be re-analyzed.

- To indicate a coordinate adjective. **Use:** Billing and Customer Care offers a robust, scalable solution for enterprises.

- To separate two independent clauses joined by a coordinating conjunction. **Use:** Click **Add another**, or click **OK** to close the window.

A comma is not necessary if the introductory clause of a sentence is short, as in the following sentence: Select **Yes** and click **OK**.
Bibliography
